



TUGAS AKHIR - TE 141599

PENERAPAN SISTEM PENYEIMBANG OTOMATIS PADA MINI SEGWAY RODA DUA

Civic Duta Mahendra
NRP 2213100047

Dosen Pembimbing
Dr. Ir. Djoko Purwanto, M.Eng.
Suwito, ST.,MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - TE 141599

PENERAPAN SISTEM PENYEIMBANG OTOMATIS PADA MINI SEGWAY RODA DUA

Civic Duta Mahendra
NRP 2213100047

Dosen Pembimbing
Dr. Ir. Djoko Purwanto, M.Eng.
Suwito, ST.,MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TE 141599

IMPLEMENTATION OF SELF-BALANCING SYSTEM ON TWO-WHEELED MINI SEGWAY

Civic Duta Mahendra
NRP 2213100047

Supervisor
Dr. Ir. Djoko Purwanto, M.Eng.
Suwito, ST.,MT.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology
Sepuluh Nopember Institute of Technology
Surabaya 2017

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Penerapan Sistem Penyeimbang Otomatis Pada *Mini Segway Roda Dua***” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 19 Juli 2017

Civic Duta Mahendra
2213100047

**PENERAPAN SISTEM PENYEIMBANG OTOMATIS PADA
MINI SEGWAY RODA DUA**

TUGAS AKHIR

**Diajukan untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada**

**Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Ir. Djoko Purwanto, M.Eng

NIP. 196512111990021002

Suwito, ST., MT.

NIP. 198101052005011004



JULI, 2017

PENERAPAN SISTEM PENYEIMBANG OTOMATIS PADA MINI SEGWAY RODA DUA

Nama : Civic Duta Mahendra
Pembimbing I : Dr. Ir. Djoko Purwanto, M.Eng.
Pembimbing II : Suwito, ST., MT.

ABSTRAK

Hasil penelitian menunjukkan bahwa miniatur *balancing robot* dapat digunakan sebagai prototipe *single human transporter*. Salah satu prototipe *single human transporter* adalah *Mini Segway*. *Mini segway* adalah skuter listrik *self-balancing* otomatis roda dua yang dapat dikontrol dengan kaki yang dirancang dengan memadukan teknik mekatronika, teknik kontrol, dan *software*. *Mini Segway* terdiri atas 3 komponen utama yaitu sensor *gyroscope* dan *accelerometer*, sistem kontrol *self balancing*, dan mekanik *self balancing* sebagai *actuator* berupa *brushless* DC motor (BLDC). Dalam tugas akhir ini, digunakan kombinasi dari *gyroscope* dan *accelerometer* yang terdapat pada IMU (*Inertial Measurement Unit*) sebagai sensor penyeimbang. Pergerakan terjadi pada sumbu-x (sudut *pitch*) dengan arah kedepan dan kebelakang. Sistem kontrol berfungsi untuk menyeimbangkan dan menjaga *chassis* pada sudut *pitch* tetap di posisi $\pm 0^\circ$ (tegak lurus). Sistem kontrol yang digunakan adalah *PID controller*. Hasil pengujian yang dilakukan pada tugas akhir ini adalah prototipe desain sistem penyeimbang otomatis pada *mini segway* roda dua belum mampu menjaga kestabilan pada sudut 0° terhadap *chassis* dikarenakan respon motor untuk pergantian arah gerak maju-mundur yang menggunakan *controller* BLDC sebagai penggerak motor terlalu lamban dan terdapat *death zone* yang terlalu besar. Namun dapat melakukan pergerakan (*steering*) sesuai dengan perintah pengguna baik bergerak maju, mundur, belok kanan, belok kiri, dan berotasi. Nilai parameter PID yang digunakan untuk pergerakan pada *Mini Segway* dalam kondisi diberikan beban pengendara sebesar 1.5 Kg adalah dengan nilai $K_p=11$, $K_i=0$ dan $K_d=3$. Selain itu diketahui bahwa karakteristik sistem kontrol pada *mini segway* adalah tidak linier.

Kata kunci : *Mini Segway*, *PID controller*, *pitch* dan *yaw*, *gyroscope* dan *accelerometer*

Halaman ini sengaja dikosongkan

IMPLEMENTATION OF SELF – BALANCING SYSTEM ON TWO – WHEELED MINI SEGWAY

Name : Civic Duta Mahendra
Supervisor : Dr. Ir. Djoko Purwanto, M.Eng.
Co-Supervisor : Suwito, ST., MT.

ABSTRACT

The result show that miniature balancing robot can be used as prototype of single human transporter. One prototype single human transporter is the Mini Segway. The mini segway is a two – wheeled self – balancing electric motor which is designed by combining mechatronic techniques, control techniques, and software. Mini segway consists of 3 main components : gyroscope and accelerometer sensor, self balancing control system, and self balancing mechanical as actuator in the form of brushless DC motor (BLDC). In this final project, used combination of gyroscope and accelerometer contained in IMU (Inertial Measurement Unit) as balancing sensor. The movement occurs on the x-axis (pitch angle) with the forward and backward directions. The control system serves to balance and keep the chassis at a fixed pitch angle in the $\pm 0^\circ$ position (perpendicular). The control system used is PID controller. The result of testing done in this final project is prototype design of automatic balancing system on mini segway two wheel has not been able to keep stability at angle 0° to chassis because of motor response to change forward-backward motion using BLDC controller as motor driver too slow and there is death zone which is too big. But it can perform the movement (steering) in accordance with the user command either move forward, backward, turn right, turn left, and rotate. The parameter value of PID used for the movement on the Mini Segway under the conditions given the rider load of 1.5 Kg is the value $K_p = 11$, $K_i = 0$ and $K_d = 3$. Also note that the control system characteristics on the mini segway is not linear.

Key words : Mini Segway, PID controller, pitch and yaw, gyroscope and accelerometer

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas kasih dan rahmat-Nya penulis dapat menyelesaikan Tuga Akhir ini dengan judul :

Penerapan Sistem Penyeimbang Otomatis Pada Mini Segway Roda Dua

Tugas akhir ini merupakan persyaratan dalam menyelesaikan pendidikan program Strata-satu di Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya.

Tugas akhir ini dibuat berdasarkan teori – teori yang didapat selama mengikuti perkuliahan, berbagai literatur penunjang dan pengarahan dosen pembimbing dari awal hingga akhir pengerjaan Tuga Akhir ini.

Pada kesempatan ini, penulis ingin berterima kasih kepada pihak – pihak yang telah membantu dalam pembuatan tugas akhir ini, khususnya kepada :

1. Orang tua serta seluruh keluarga yang memberikan dukungan baik moril maupun materiil.
2. Dr. Ir. Djoko Purwanto, M.Eng. selaku dosen pembimbing 1 atas bimbingan dan arahan selama penulis mengerjakan tugas akhir ini.
3. Suwito, ST. MT. selaku dosen pembimbing 2 atas bimbingan dan arahan selama penulis mengerjakan tugas akhir ini.
4. Dr. Muhammad Rivai, ST. MT. selaku Kepala Laboratorium Elektronika Industri.
5. Dr. Eng. Ardyono Priyadi, ST. M.Eng. selaku Kepala Departemen Teknik Elektro ITS Surabaya.
6. Seluruh dosen Teknik Elektro, khususnya dosen bidang studi elektronika.
7. Teman – teman e53, senior e52 serta junior e54 yang tidak dapat disebutkan satu – persatu, telah banyak membantu proses pengerjaan tugas akhir ini.

Penulis sadar bahwa Tugas Akhir ini jauh dari sempurna dan masih banyak hal yang dapat diperbaiki. Kritik dan saran yang bersifat membangun diharapkan dari semua pihak untuk pengembangan lebih lanjut, khususnya untuk pengembangan *self – balancing vehicle*.

Surabaya, 19 Juli 2017

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

ABSTRAK.....	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	xiii
 BAB I PENDAHULUAN	 1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Metodologi	3
1.6 Sistematika Penulisan	4
1.7 Relevansi dan Manfaat	5
 BAB II TINJAUAN PUSTAKA DAN TEORI PENUNJANG	 7
2.1 Perbedaan Antara <i>Segway</i> dan <i>Mini Segway</i>	7
2.2 <i>Inverted Pendulum</i>	8
2.3 <i>PID Controller</i>	10
2.3.1 <i>Controller Proportional (P)</i>	10
2.3.2 <i>Controller Integral (I)</i>	11
2.3.3 <i>Controller Derivative (D)</i>	11
2.3.4 Blok Diagram, Flowchart dan Persamaan Kontrol <i>PID</i>	11
2.4 <i>Pulse Width Modulation</i>	13
2.5 Komunikasi I2C.....	14
2.6 Komunikasi Serial	16
2.7 Komponen Penyusun <i>Mini Segway</i>	19
2.7.1 IMU MPU6050	19
2.7.1.1 Fitur <i>Gyroscope</i>	20
2.7.1.2 Fitur <i>Accelerometer</i>	20
2.7.1.3 Fitur Tambahan.....	21
2.7.1.4 <i>Motion Processing</i>	22
2.7.1.5 <i>Clocking</i>	22
2.7.2 <i>Arduino Nano</i>	23
2.7.3 <i>Maple Mini</i>	24

2.7.4 <i>Buck Converter</i>	25
2.7.5 <i>Controller BLDC 350W Full Fitur</i>	26
2.7.6 <i>BLDC Motor</i>	28
2.7.7 <i>Baterai Pack</i>	30
2.7.8 <i>Saklar</i>	30
BAB III PERANCANGAN SISTEM	31
3.1 <i>Desain Hardware Mini Segway</i>	33
3.1.1 <i>Board Sensor</i>	34
3.1.2 <i>Main Board</i>	36
3.2 <i>Desain Software Mini Segway</i>	42
3.3 <i>Perancangan Pin Out Yang Digunakan Pada Arduino</i>	45
3.4 <i>Perancangan Pin Out Yang Digunakan Pada Maple Mini</i>	46
BAB IV PENGUJIAN	47
4.1 <i>Kalibrasi Sensor IMU MPU6050</i>	47
4.2 <i>Pengujian Sensor IMU MPU6050</i>	48
4.3 <i>Pengiriman Data Sudut</i>	49
4.4 <i>Penerimaan Data Sudut</i>	50
4.5 <i>Pengolahan Data Sudut</i>	51
4.6 <i>Pengujian Controller BLDC</i>	61
4.7 <i>Pengujian Motor BLDC</i>	62
BAB V PENUTUP	65
5.1 <i>Kesimpulan</i>	65
5.2 <i>Saran</i>	65
DAFTAR PUSTAKA	67
LAMPIRAN	69
BIODATA PENULIS	71

DAFTAR GAMBAR

Gambar 2.1 <i>Segway</i>	7
Gambar 2.2 <i>Mini Segway</i>	8
Gambar 2.3 Model Sistem JOE	9
Gambar 2.4 Ilustrasi Pergerakan	9
Gambar 2.5 Ilustrasi Sistem	10
Gambar 2.6 Blok Diagram Kontrol PID	11
Gambar 2.7 <i>Duty Cycle</i> Pada PWM	14
Gambar 2.8 Kondisi Sinyal <i>Start</i> dan <i>Stop</i>	15
Gambar 2.9 <i>Transfer</i> Bit Pada I2C Bus	15
Gambar 2.10 Pengalamatan I2C	16
Gambar 2.11 Komunikasi Serial	16
Gambar 2.12 <i>Serial Frame</i>	17
Gambar 2.13 Paket Data Serial ‘OK’	18
Gambar 2.14 IMU MPU6050	19
Gambar 2.15 <i>Microcontroller</i> Arduino Nano	24
Gambar 2.16 <i>Microcontroller</i> Maple Mini	25
Gambar 2.17 <i>Buck Converter</i>	26
Gambar 2.18 <i>Controller</i> BLDC 350W <i>Full</i> Fitur	27
Gambar 2.19 Motor <i>Brushless</i>	28
Gambar 2.20 Kabel Motor <i>Brushless</i>	29
Gambar 2.21 Sinyal 3 Fasa Dengan <i>Hall Sensor</i>	29
Gambar 2.22 Baterai <i>Pack</i>	30
Gambar 2.23 Saklar	30
Gambar 3.1 Desain Keseluruhan Perancangan <i>Hardware</i>	31
Gambar 3.2 Blok Diagram Sistem Kontrol	31
Gambar 3.3 Blok Diagram Sistem	32
Gambar 3.4 Desain <i>Hardware</i> <i>Mini Segway</i> Tampak Atas	33
Gambar 3.5 Desain <i>Hardware</i> <i>Mini Segway</i> Tampak Bawah	33
Gambar 3.6 <i>Board</i> Sensor	34
Gambar 3.7 Skematik Rangkaian Keseluruhan <i>Board</i> Sensor	35
Gambar 3.8 <i>Board</i> PCB Keseluruhan Pada <i>Board</i> Sensor	36
Gambar 3.9 <i>Main Board</i>	37
Gambar 3.10 Skematik Rangkaian Keseluruhan <i>Main Board</i>	38
Gambar 3.11 <i>Board</i> PCB Keseluruhan Pada <i>Main Board</i>	39
Gambar 3.12 Skematik Rangkaian Penstabil Tegangan	40
Gambar 3.13 Skematik Rangkaian Komunikasi Serial	40
Gambar 3.14 Skematik Rangkaian Arduino Bagian Atas	41

Gambar 3.15 Skematik Rangkaian Arduino Bagian Bawah	41
Gambar 3.16 Skematik Rangkaian Maple Mini	42
Gambar 3.17 <i>Flowchart</i> Untuk Mendapatkan Integral	42
Gambar 3.18 <i>Flowchart</i> Untuk Mendapatkan Derivatif	43
Gambar 3.19 <i>Flowchart</i> Persamaan PID.....	44
Gambar 3.20 Algoritma Penghitungan Nilai <i>Output</i> PID	44
Gambar 3.21 <i>Flowchart</i> Persamaan PID.....	44
Gambar 3.22 Pin <i>Out</i> Arduino Nano.....	45
Gambar 3.23 Pin <i>Out</i> Maple Mini.....	46
Gambar 4.1 Kalibrasi Sensor IMU MPU6050 Dengan Aplikasi <i>Elixir</i> Pada <i>Handphone</i>	47
Gambar 4.2 Hasil Kalibrasi Sensor IMU MPU6050.....	48
Gambar 4.3 Data Hasil Pengujian Sensor IMU MPU6050.....	49
Gambar 4.4 Program Pengiriman Data Sudut	49
Gambar 4.5 Pengiriman Data Sudut Per Byte	50
Gambar 4.6 Penerimaan Data Sudut	51
Gambar 4.7 Pengujian Penentuan Batas Minimal Nilai K_p	52
Gambar 4.8 Grafik Respon Sudut Dengan Nilai $K_p = 1$	52
Gambar 4.9 Grafik Respon Sudut Dengan Nilai $K_p = 2$	53
Gambar 4.10 Grafik Respon Sudut Dengan Nilai $K_p = 3$	53
Gambar 4.11 Grafik Respon Sudut Dengan Nilai $K_p = 4$	54
Gambar 4.12 Grafik Respon Sudut Dengan Nilai $K_p = 5$	54
Gambar 4.13 Grafik Respon Sudut Dengan Nilai $K_p = 6$	55
Gambar 4.14 Grafik Respon Sudut Dengan Nilai $K_p = 7$	55
Gambar 4.15 Grafik Respon Sudut Dengan Nilai $K_p = 8$	56
Gambar 4.16 Grafik Respon Sudut Dengan Nilai $K_p = 9$	56
Gambar 4.17 Grafik Respon Sudut Dengan Nilai $K_p = 10$	57
Gambar 4.18 Grafik Respon Sudut Dengan Nilai $K_p = 11$	57
Gambar 4.19 Pengujian Keseimbangan <i>Mini Segway</i>	58
Gambar 4.20 Grafik Respon Sudut Dengan Nilai $K_p = 11$ dan $K_d = 1$	58
Gambar 4.21 Grafik Respon Sudut Dengan Nilai $K_p = 11$ dan $K_d = 2$	59
Gambar 4.22 Grafik Respon Sudut Dengan Nilai $K_p = 11$ dan $K_d = 3$	59
Gambar 4.23 Grafik Respon Sudut Dengan Nilai $K_p = 11$, $K_d = 3$ dan Max PWM = 50	60
Gambar 4.24 Grafik Respon Sudut Dengan Nilai $K_p = 11$, $K_d = 3$ dan Max PWM = 45	60

Gambar 4.25 Rangkaian Penaik Tegangan PWM	62
Gambar 4.26 Sinyal <i>Hall Sensor</i> Kabel U (atas) dan kabel V (bawah)	63
Gambar 4.27 Sinyal <i>Hall Sensor</i> Kabel V (atas) dan kabel W (bawah)	63
Gambar 4.28 Sinyal <i>Hall Sensor</i> Kabel W (atas) dan kabel U (bawah)	64

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1 <i>State Variable</i> Yang Diubah.....	9
Tabel 2.2 Efek Dari Setiap <i>Controller</i> PID Pada Sistem.....	13
Tabel 2.3 Tabel Kode Bilangan ASCII	19
Tabel 3.1 Tabel Komponen Pada Rangkaian <i>Board</i> Sensor.....	35
Tabel 3.2 Tabel Denah Penetapan Kabel Pada <i>Main Board</i>	39
Tabel 3.3 Tabel Komponen Pada Rangkaian Penstabil Tegangan ...	40
Tabel 3.4 Tabel Konfigurasi Pin <i>Out</i> Pada Arduino Nano Dengan MPU6050.....	45
Tabel 3.5 Tabel Konfigurasi Pin <i>Out</i> Maple Mini Dengan Arduino Nano.....	46
Tabel 4.1 Data <i>Offset</i> Masing – Masing Sensor	48
Tabel 4.2 Perbandingan Frekuensi PWM Pada Gerak Motor	62

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Permasalahan terbesar didunia adalah permasalahan energi, dimana energi yang berasal dari minyak bumi mengalami krisis yang cukup besar. Dengan berkurangnya bahan bakar minyak dan kebutuhan konsumen yang meningkat mengakibatkan nilai jual yang melambung tinggi. Bahan bakar minyak bumi banyak digunakan pada transportasi. Dan apabila ditinjau dari perhitungan, penggunaan energi listrik masih jauh lebih murah dan mudah didapatkan dibandingkan dengan energi minyak bumi. Oleh karena itu transportasi berenergi listrik yang murah dari segi bahan bakar dan handal muncul sebagai salah satu solusi dalam krisis energi yang terjadi. Penelitian dan pengembangan *Self Balancing Robot* pun dilakukan. Sejak Dean Kamen, 2001 telah mendeklarasikan SEGWAY sebagai “*The first Self-balancing, electric powered transportation device*” kemudian diikuti oleh penelitian – penelitian sejenis di berbagai negara. Hal ini tentunya dipicu oleh pengembangan teknologi elektronika, robotik serta IT yang cukup pesat di seluruh dunia. *Self-balancing robot* merupakan robot yang memiliki prinsip kerja yang menerapkan sistem *inverted pendulum*.

Dalam kajian pustaka yang telah dilakukan ada beberapa penelitian serupa dengan beberapa perbedaan metode kontrol dan penggunaan mikrokontroller dan komponen pendukungnya. Penelitian tentang desain kontrol dari *two-wheeled self-balancing robot* didasarkan pada ADRC (*Active Disturbances Rejection Controller*). Dijelaskan bahwa *disturbances rejection* adalah masalah utama dari desain kontrol *two-wheeled self-balancing*. Sehingga diusulkan skema kontrol ganda *double-loop* berdasarkan ADRC untuk menerapkan kontrol *stable upright control* pada robot. Dalam skema ini, algoritma kontrol PID digunakan sebagai *inner-loop controller* untuk mengeksekusi kecepatan *close loop* dari motor, dan teknik ADRC diterapkan untuk merancang pengontrol sudut *pitch* robot dimana desain parameter dioptimalkan dari ESO yang digunakan untuk meningkatkan kinerja kontroler. Hasil ekstensif simulasi dan percobaan menunjukkan bahwa skema kontrol yang diusulkan ini jika dibandingkan dengan kontrol tradisional berupa *double-loop PID control* memiliki ketahanan yang lebih kuat

terhadap perubahan beban dan gangguan eksternal. Penelitian lain yaitu menggunakan PID *Controller* namun pada *one-wheeled self-balancing vehicle*. *PENS-Wheel* adalah kendaraan listrik yang menggunakan satu roda dan mampu menyeimbangkan diri sendiri sehingga pengendara tidak jatuh ke depan atau ke belakang saat naik. Kendaraan ini menggunakan satu *brushless* DC motor sebagai *actuator* yang mampu memutar dikedua arah simetris. Kendaraan ini menggunakan kombinasi *accelerometer* dan *gyroscope* yang ada pada IMU (*Inertial Measurement Unit*) untuk sensor penyeimbangan. Kontrol pergerakan pada kendaraan hanya terjadi pada sumbu-x (sumbu *pitch*), arah pergerakan kedepan dan kebelakang. Algoritma kontrol PID (*Proportional Integral Derivative*) digunakan untuk menjaga keseimbangan dan pergerakan kendaraan. Dari simulasi dan aplikasi dalam prakteknya, penggunaan kontrol PID mampu mengemudikan kendaraan dalam menjaga kondisi keseimbangan dengan batas sudut pada permukaan datar $\pm 10^\circ$, jalan yang bergelombang, dan jalan dengan kecuraman 15° .

Setelah mengulas beberapa penelitian sebelumnya, maka dengan menggabungkan *Intelligent Control System* (ICS) berupa penggunaan metode *PID Controller* yang akan diimplementasikan ke dalam *microcontroller* dengan input berupa *error* sudut, sensor berupa *gyrometer* dan *accelerometer* (IMU MPU6050) dan *actuator* berupa *brushless* DC motor (BLDC). Diharapkan prototipe kendaraan *Self Balancing Robot* (SBR) *mini segway* dapat menjadi terobosan kendaraan masa depan dan didapatkan metode sistem penyeimbangan dengan hasil yang baik.

1.2 Perumusan Masalah

Permasalahan yang dibahas dalam tugas akhir ini adalah :

1. Penentuan pergerakan motor terhadap perubahan sudut *pitch* pada *chassis mini segway*.
2. Penentuan stabilitas penyeimbang diri secara otomatis pada *mini segway*.

1.3 Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah :

1. Penyeimbangan dilakukan ketika *mini segway* dikendarai.
2. *Mini segway* dijalankan pada landasan yang datar.
3. Hanya digunakan oleh pengendara dengan bobot yang sama.

1.4 Tujuan

Tujuan yang ingin dicapai dalam perancangan ini adalah :

1. Motor berputar kedepan ketika sudut *pitch* pada *chassis* bernilai positif dan berputar kebelakang ketika sudut *pitch* pada *chassis* bernilai negatif.
2. Sistem kontrol dapat menentukan posisi kestabilan dan menjalankan motor DC *Brushless* (BLDC) sesuai dengan output kontrol yang menyebabkan perubahan sudut *pitch* pada *chassis* sehingga perubahan sudut *yaw* pada *mini segway* menjadi 0°.

1.5 Metodologi

Langkah – langkah yang dikerjakan pada tugas akhir ini adalah sebagai berikut :

1. Studi Literatur

Tahap ini meliputi pengumpulan dasar teori yang dapat menjadi acuan tugas akhir. Dasar teori akan diambil dari buku – buku, jurnal yang telah dipublikasi, *proceeding* dan artikel maupun forum hasil diskusi dari internet.

2. Perancangan Hardware

Tahapan ini berupa perancangan *hardware*, secara umum meliputi desain prototipe *mini segway*. Sensor menggunakan dua sensor IMU (*Inertial Measurement Unit*) MPU6050 yang didalamnya terdapat *gyrometer* dan *accelerometer*, diletakkan pada masing – masing sisi pijakan *mini segway*. *Microcontroller* menggunakan Arduino Nano dan Maple Mini. Baterai menggunakan baterai *Lithium-Ion* berkapasitas 36V 4400mAh. Motor menggunakan motor jenis *brushless* dengan daya motor 350W.

3. Perancangan Software

Tahapan ini berupa perancangan *software* program pengambilan dan pengiriman data sudut, dan algoritma program penyeimbang untuk prototipe *mini segway*. *Software* menggunakan Arduino-IDE untuk pemrograman pengambilan

data sudut dari sensor IMU, dan Maple-IDE untuk pemrograman kontrol PID.

4. Pengujian Sistem

Tahap ini berupa pengujian prototipe yang dilakukan untuk menentukan seberapa besar *error* yang dihasilkan dan *feedback* pergerakan *mini segway*. Pengujian sistem pertama dilakukan untuk mengetahui metode kontrol PID sudah berjalan sesuai dengan yang diinginkan atau belum dengan pengaturan pada parameter nilai K_p , K_i dan K_d sehingga dihasilkan output sinyal PWM (*Pulse Width Modulation*) sebagai *input driver* motor yang kemudian sinyal tersebut akan mengatur pergerakan dari motor *brushless* sesuai dengan nilai *pitch* yang ada terhadap *chassis*.

5. Analisa

Tahap ini berupa analisa yang dilakukan terhadap hasil dari pengujian yang dilakukan sehingga dapat menentukan karakteristik dari *software* maupun *hardware* yang telah dibuat. Apabila karakteristik pergerakan motor dari *software* dan *hardware* masih belum sesuai maka perlu dilakukan perancangan ulang pada sistem, namun lebih ditekankan pada perancangan ulang *software*.

6. Penyusunan Laporan Tugas Akhir

Tahap penulisan laporan tugas akhir adalah tahapan terakhir dari proses pengerjaan tugas akhir ini. Laporan tugas akhir berisi seluruh hal yang berkaitan dengan tugas akhir yang telah dikerjakan yaitu meliputi pendahuluan, tinjauan pustaka, teori penunjang, perancangan sistem, pengujian, dan penutup.

1.6 Sistematika Penulisan

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika penulisan sebagai berikut :

➤ **BAB I : PENDAHULUAN**

Bab ini meliputi penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi.

➤ **BAB II : TINJAUAN PUSTAKA DAN TEORI PENUNJANG**

Bab ini menjelaskan tentang teori penunjang dan *literature* yang dibutuhkan dan berguna dalam pengerjaan tugas akhir ini.

Dasar teori penunjang yang meliputi perbedaan antara *segway* dan *mini segway*, *inverted pendulum*, *PID controller*, dan *pulse width modulation*.

➤ **BAB III : PERANCANGAN SISTEM**

Bab ini menjelaskan tentang perancangan sistem baik perangkat keras (*hardware*) maupun perangkat lunak (*software*) untuk sistem penyeimbang otomatis pada *mini segway* roda dua.

➤ **BAB IV : PENGUJIAN**

Pada bab ini menjelaskan hasil uji coba sistem beserta analisisnya.

➤ **BAB V : PENUTUP**

Bagian ini merupakan bagian akhir yang berisikan kesimpulan yang diperoleh dari pembuatan Tugas Akhir ini, serta saran – saran untuk pengembangan lebih lanjut.

1.7 Relevansi dan Manfaat

Hasil yang diharapkan dari tugas akhir ini diharapkan mampu meringankan pekerjaan manusia pada bidang transportasi. Mengurangi polusi dan sebagai bentuk aplikasi penggunaan energi terbarukan karena bertenaga listrik. Sebagai pengembangan dalam dunia riset tentang *self-balancing vehicle* yang diharapkan dapat digunakan pada transportasi dikemudian hari. Ketika teknologi *self-balancing* diterapkan pada transportasi umum diharapkan dapat mengurangi kecelakaan berupa jatuhnya dari kendaraan yang diakibatkan oleh kelalaian pengendara.

Halaman ini sengaja dikosongkan

BAB II

TINJAUAN PUSTAKA DAN TEORI PENUNJANG

Dalam bab ini menjelaskan tentang teori penunjang yang berhubungan dengan keseluruhan sistem yang akan dibuat pada tugas akhir ini. Sedangkan tinjauan pustaka dalam bab ini menjelaskan tentang sistem – sistem yang berhubungan dengan tugas akhir ini dan pernah diimplementasikan oleh penulis – penulis sebelumnya.

2.1 Perbedaan Antara *Segway* dan *Mini Segway*

Segway adalah *The first Self-balancing, electric powered transportation device*. *Mini segway* adalah pengembangan dari *segway* itu sendiri. *Segway* memiliki bentuk dan ukuran yang lebih besar, sedangkan *mini segway* memiliki ukuran yang lebih kecil. Dari desainnya, *segway* memiliki pegangan tangan, sedangkan *mini segway* tidak ada. Dan yang membedakan lagi adalah *mini segway* dapat berputar 360° ditempat, sedangkan *segway* tidak. Hal ini karena *segway* bekerja dengan kemiringan sudut oleh tubuh, sedangkan *mini segway* bekerja dengan kemiringan sudut kaki.



Gambar 2.1 *Segway*

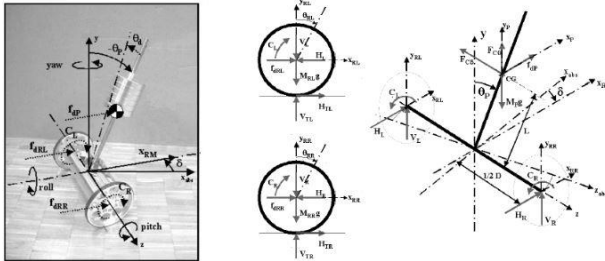


Gambar 2.2 *Mini Segway*

2.2 *Inverted Pendulum*

Ide dasar untuk membuat robot beroda dua dapat setimbang adalah sangat mudah yaitu dengan cara mengendalikan roda searah dengan arah jatuhnya bagian atas sebuah robot. Apabila proses tersebut dapat terlaksana maka robot tersebut dapat setimbang. Secara praktis membutuhkan dua sensor sebagai umpan balik yaitu sensor kemiringan atau sudut terhadap gaya gravitasi dan sensor posisi.

Dalam penelitian *JOE: A Mobile, Inverted Pendulum* menggunakan 3 *Degree of Freedom* (DoF) (Grasser dkk, 2001), sistem dapat berputar terhadap sumbu z (*pitch*), pergerakannya dideskripsikan sebagai posisi P (θ_p) dan ω_p (kecepatan sudut/*angular velocity*). Pergerakan linier dideskripsikan dalam translasi (x_{RM}) dan kecepatan linier (v_{RM})[1]. Perputaran sumbu vertikal (*yaw*) dikarenakan putaran roda. Putaran tersebut dapat dideskripsikan sebagai δ dan kecepatan sudut $d(\delta)$. Model sistem JOE dapat digambarkan sebagai berikut :

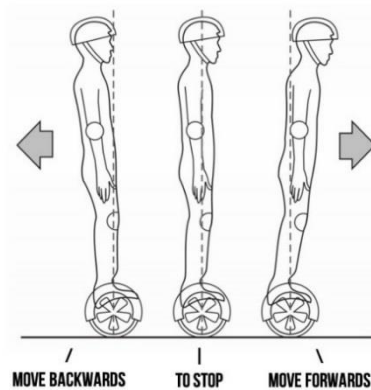


Gambar 2.3 Model Sistem JOE

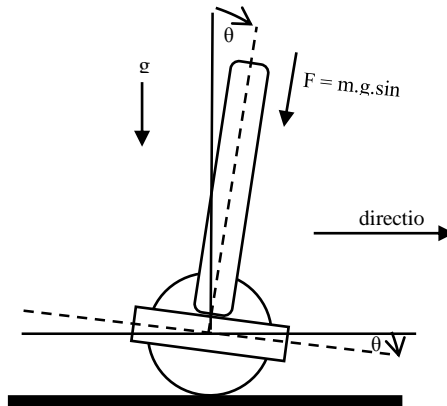
Berikut ini 6 *state space* variabel yang dapat diubah antara lain :

Tabel 2.1 *State Variable* Yang Diubah

X_{RM}	<i>straight line position</i>	[m]
V_{RM}	<i>straight line speed</i>	[m/s]
θ_p	<i>pitch angle</i>	[rad]
ω_p	<i>pitch rate</i>	[rad/s]
δ	<i>yaw angle</i>	[rad]
$d(\delta)$	<i>yaw rate</i>	[rad/s]



Gambar 2.4 Ilustrasi Pergerakan



Gambar 2.5 Ilustrasi Sistem

Keterangan :

- θ : sudut (rad)
- θ_w : percepatan sudut motor (rad/s)
- m : massa pengendara (kg)
- g : percepatan gravitasi (9.8 m/s^2)
- F : gaya (N)
- direction : arah gerak *mini segway*

2.3 PID Controller

PID (*Proportional-Integral-Derivative*) merupakan *controller* untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik (*feedback*) pada sistem tersebut. Komponen kontrol PID ini terdiri dari tiga jenis meliputi Proporsional, Integratif dan Derivatif. Konfigurasi tidak bisa Integral atau Derivatif sendiri. Konfigurasi yang mungkin tergantung *plan* yang diinginkan (P, PI, PD, PID).

2.3.1 Controller Proportional (P)

Pada sistem, *controller proportional* dapat menambah/mengurangi kestabilan, memperbaiki respon transien khususnya : *rise time* dan *settling time*, dan

mengurangi *error steady state* namun tidak menghilangkannya. Untuk menghilangkan *error steady state* dibutuhkan K_p (konstanta *proportional*) yang besar, yang akan membuat sistem lebih tidak stabil dan menyebabkan respon berosilasi. *Controller* proportional memberi pengaruh langsung yang sebanding pada *error*. Semakin besar error, semakin besar sinyal kendali yang dihasilkan *controller*.

2.3.2 *Controller Integral (I)*

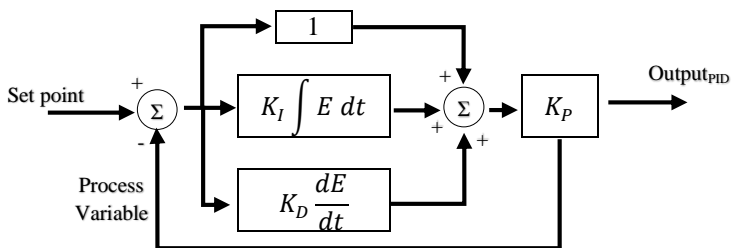
Pada sistem, *controller integral* dapat menghilangkan *error steady state*, namun respon lebih lambat jika dibandingkan dengan P, dapat menambah kestabilan karena menambah orde pada sistem. K_i (konstanta *integral*) yang besar akan mempercepat hilangnya *offset*, tetapi semakin besar nilai konstanta K_i akan mengakibatkan peningkatan osilasi dari sinyal keluaran *controller*.

2.3.3 *Controller Derivative (D)*

Pada sistem, *controller derivative* dapat memberikan efek redaman pada sistem yang berosilasi sehingga bisa memperbesar pemberian nilai K_p . Selain itu dapat memperbaiki respon transien, karena memberikan aksi saat ada perubahan *error*. Dengan meningkatkan nilai K_d dapat meningkatkan stabilitas sistem dan mengurangi *overshoot*. D hanya berubah saat ada perubahan *error*, sehingga saat ada *error* statis D tidak beraksi. Sehingga D tidak boleh digunakan sendiri.

2.3.4 **Blok Diagram, Flowchart dan Persamaan Kontrol PID**

PID Blok Diagram dapat dilihat pada gambar dibawah ini :



Gambar 2.6 Blok Diagram Kontrol PID

Adapun persamaan kontrol PID adalah :

$$Output_{PID} = K_p \left[E + K_I \int E dt + K_D \frac{dE}{dt} \right]$$

Atau :

$$Output_{PID} = K_p \left[E + K_I \sum E \Delta t + K_D \frac{\Delta E}{\Delta t} \right]$$

Keterangan :

$Output_{PID}$	= output dari PID kontrol
K_p	= konstanta Proporsional
K_I	= konstanta Integral
K_D	= konstanta Derivatif
E	= <i>error</i> (selisih antara set point dengan level aktual)
Δt	= <i>time sampling</i>

Efek dari setiap kontrol proportional, integral dan derivatif pada sistem *close loop* disimpulkan pada Tabel 2.2. Setiap kekurangan dan kelebihan dari masing – masing kontrol P, I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontrol PID (P+I+D). Secara keseluruhan kontrol PID bertujuan untuk :

1. Mempercepat reaksi sebuah sistem mencapai *set point*-nya. Menaikkan K_p akan menghasilkan respon yang kaku dan cepat.
2. Menghilangkan *offset*. Menaikkan K_i akan mengurangi waktu yang dibutuhkan untuk mencapai *error* nol.
3. Menghasilkan perubahan awal yang besar. Menaikkan K_d akan mengurangi *overshoot*.

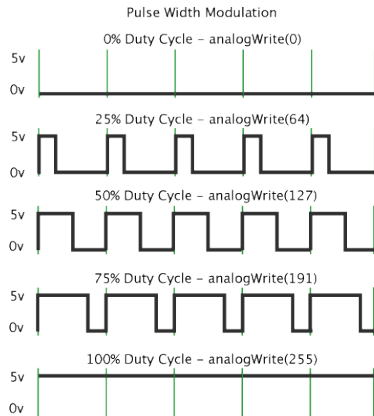
Tabel 2.2 Efek Dari Setiap *Controller* PID Pada Sistem

<i>Respon Close Loop</i>	<i>Rise Time</i>	<i>Overshoot</i>	<i>Settling Time</i>	<i>Error Steady State</i>
<i>Proportional</i>	Menurunkan	Meningkatkan	Perubahan Kecil	Menurunkan/mengurangi
<i>Integral</i>	Menurunkan	Meningkatkan	Meningkatkan	Mengeliminasi
<i>Derivative</i>	Perubahan Kecil	Menurunkan	Menurunkan	Perubahan Kecil

2.4 Pulse Width Modulation

Dasar *pulse width modulation* (PWM) secara luas digunakan di dalam aplikasi elektronika daya untuk pengaturan pengkonversian daya (DC/DC, DC/AC, dll)[2]. Secara sederhana, PWM merupakan sinyal dengan lebar pulsa yang bernilai “HIGH” dalam satu periode mewakili tegangan DC, tergantung nilai *duty cycle*. *Duty cycle* merupakan perbandingan lama waktu sinyal bernilai “HIGH” dengan satu periode. Gambar 2.8 menggambarkan *duty cycle* dari PWM.

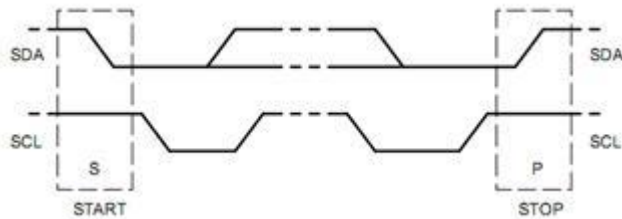
Selain digunakan dalam pengaturan pengkonversian daya, ada beberapa aplikasi lain dari PWM. Contoh aplikasi umum lainnya adalah pengendalian kecepatan motor DC, pengendalian motor servo, pengaturan nyala LED dan lain sebagainya.



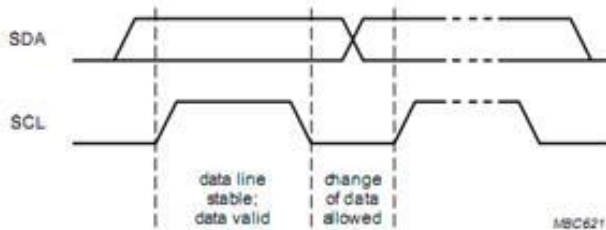
Gambar 2.7 Duty Cycle Pada PWM

2.5 Komunikasi I2C

Inter Integrated Circuit atau sering disebut I2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data. Sistem I2C terdiri dari saluran SCL (*Serial Clock*) dan SDA (*Serial Data*) yang membawa informasi data antara I2C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai *Master* dan *Slave*. *Master* adalah piranti yang memulai *transfer* data pada I2C Bus dengan membentuk sinyal *Start*, dan mengakhiri *transfer* data dengan membentuk sinyal *Stop*, dan membangkitkan sinyal *clock*. *Slave* adalah piranti yang dialamati *Master*. Sinyal *Start* merupakan sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “1” menjadi “0” pada saat SCL “1”. Sinyal *Stop* merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “0” menjadi “1” pada saat SCL “1”. Kondisi sinyal *Start* dan sinyal *Stop* seperti pada Gambar 2.8.



Gambar 2.8 Kondisi Sinyal *Start* dan *Stop*

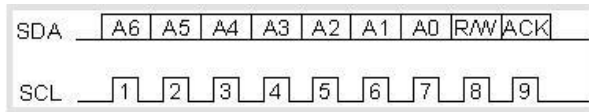


Gambar 2.9 *Transfer Bit* Pada I2C Bus

Dalam melakukan *transfer* data pada I2C bus, dilakukan tata cara yang telah ditetapkan yaitu :

- *Transfer* data hanya dapat dilakukan ketika Bus tidak dalam keadaan sibuk.
- Selama proses *transfer* data, keadaan data pada SDA harus stabil selama SCL dalam keadaan tinggi. Keadaan perubahan “1” atau “0” pada SDA hanya dapat dilakukan selama SCL dalam keadaan rendah. Jika terjadi perubahan keadaan SDA pada saat SCL dalam keadaan tinggi, maka perubahan itu dianggap sebagai sinyal *Start* atau sinyal *Stop*.

Dalam komunikasi I2C terdapat mode pengalamatan dari *Master* dan *Slave*. Mode pengalamatan memungkinkan untuk mendapatkan data yang benar sesuai dengan data yang tersimpan didalam komponen tersebut. Mode pengalamatan terdiri 7 bit + 1 bit R/W + 1 bit ACK. Jika 1 *device* terdiri dari alamat 7 bit data, maka logikanya dalam 1 bus bisa dipasang sampai dengan 127 *device*. Pengiriman data pertama dalam I2C adalah MSB (Most Significant



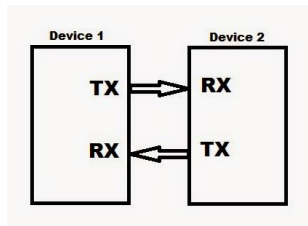
Gambar 2.10 Pengalamatan I2C

Bit) dan terakhir adalah LSB (Least Significant Bit) kemudian dilanjutkan dengan 1 bit ACK.

2.6 Komunikasi Serial

Serial berarti mengirim satu bit data dan selanjutnya diikuti oleh bit – bit data yang lain pada jalur yang sama. Komunikasi serial adalah komunikasi dengan mengirimkan data per bit secara berurutan dan bergantian. Komunikasi ini mempunyai kelebihan yaitu hanya membutuhkan satu jalur dan kabel yang sedikit. Pada prinsipnya komunikasi serial merupakan komunikasi dimana pengiriman data dilakukan per bit sehingga lebih lambat dibandingkan komunikasi paralel, atau dengan kata lain komunikasi serial merupakan salah satu metode komunikasi data dimana hanya satu bit data yang dikirim melalui seuntai kabel pada suatu waktu tertentu. Pada dasarnya komunikasi serial adalah kasus khusus komunikasi paralel dengan nilai $n = 1$. Hal ini dapat disandingkan dengan komunikasi paralel yang sesungguhnya dimana n -bit data dikirim bersamaan, dengan nilai umumnya $8 \leq n \leq 128$.

Komunikasi serial ada 2 macam, *asynchronous serial* dan *synchronous serial*. *Synchronous serial* adalah komunikasi dimana hanya satu pihak (pengirim atau penerima) yang menghasilkan clock dan mengirimkan clock tersebut bersamaan – sama dengan data. Contoh penggunaan *synchronous serial* terdapat pada transmisi data



Gambar 2.11 Komunikasi Serial

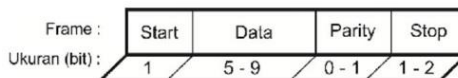
keyboard. *Asynchronous serial* adalah komunikasi dimana kedua pihak (pengirim dan penerima) masing – masing menghasilkan clock namun hanya data yang diterima, maka kedua frekuensi clock harus sama dan harus terdapat sinkronisasi. Setelah adanya sinkronisasi, pengirim akan mengirimkan datanya sesuai dengan frekuensi clock pengirim dan penerima akan membaca data sesuai dengan frekuensi clock penerima.

Terdapat beberapa parameter yang digunakan untuk membangun komunikasi secara serial, diantaranya adalah *baud rate*, paket data, *parity* bit dan *synchronization* bit. *Baud rate* mengindikasikan seberapa cepat data dikirim melalui komunikasi serial. *Baud rate* biasanya diberi satuan *bit-per-second* (bps). Bit per detik berarti bahwa berapa bit data dapat ditransfer setiap detiknya. Nilai *baud rate* dapat diatur dengan menggunakan standar kecepatan yang disediakan, diantaranya 1.200 2.400, 9.600, 19.200, 38.400, 57.600 dan 115.200 bps.

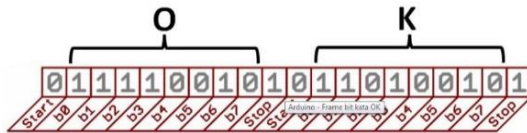
Framing data adalah bagaimana suatu rangkaian bit disusun untuk dikirim melalui suatu sistem komunikasi serial. Pada Gambar 3.11 data yang dikirim melalui komunikasi serial biasanya adalah 5 sampai 9-bit. Pada Arduino, data berukuran 8-bit (1-byte). *Start* dan *Stop* bit dikenal sebagai *synchronization* bit. *Start* dan *Stop* bit bisa berukuran 2 atau 3-bit. Sesuai dengan namanya, bit – bit ini akan mengawali dan mengakhiri paket data. *Start* bit selalu berukuran 1-bit, sedangkan *Stop* bit bisa 1 atau 2-bit.

Parity bit bersifat opsional dan dapat tidak digunakan. *Parity* bit berguna untuk data transfer yang dipengaruhi oleh *noise*. Namun, penggunaan *parity* bit dapat memperlambat kecepatan komunikasi. Penggunaan *parity* bit juga memerlukan sinkronisasi antara *transmitter* dengan *receiver*.

Melalui komunikasi serial, data akan dikirim dalam format ASCII (*American Standard Code for Information Interchange*). Misal akan mengirim data OK. Karena terdiri dari dua karakter (O dan K), maka komunikasi akan memiliki dua buah paket data. Kode



Gambar 2.12 Serial Frame



Gambar 2.13 Paket Data Serial ‘OK’

ASCII untuk ‘O’ (kapital) adalah 79 dalam desimal, atau 0100 1111 dalam 8-bit binernya, sedangkan karakter ‘K’ (kapital) adalah 75 atau 0100 1011. Data yang terkirim lebih dahulu adalah *least-significant-bit* (lsb) nya seperti pada Gambar 3.12.

Misal menggunakan *baud rate* standar yaitu 9600 bit/detik, maka setiap bit-nya memerlukan waktu selama $1/9600 = 104$ mikrodetik/bit. Satu paket data untuk satu karakter terdiri dari 10-bit (8-bit data, 1-bit start dan 1-bit stop). Dengan demikian, pengiriman satu karakter (yang terdiri dari 10-bit) akan membutuhkan waktu selama 10×104 mikrodetik = 1.040 mikrodetik = 1,04 milidetik. Pengiriman kata ‘OK’ akan membutuhkan waktu sekitar 2 milidetik pada kecepatan 9600 bps. Jika menambah sebuah parity akan mengurangi kecepatan pengiriman data.

Tabel 2.3 Tabel Kode Bilangan ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Htmi	Chr	Dec	Hx	Oct	Htmi	Chr	Dec	Hx	Oct	Htmi	Chr
0	0	000	NUL (null)	32	20	040	Space	64	40	100	@	96	60	140	#96;			
1	1	001	SOH (start of heading)	33	21	041	!	65	41	101	A	97	61	141	#97;			
2	2	002	STX (start of text)	34	22	042	"	66	42	102	B	98	62	142	#98;			
3	3	003	ETX (end of text)	35	23	043	#	67	43	103	C	99	63	143	#99;			
4	4	004	END (end of transmission)	36	24	044	\$	68	44	104	D	100	64	144	#100;			
5	5	005	ENQ (enquiry)	37	25	045	%	69	45	105	E	101	65	145	#101;			
6	6	006	ACK (acknowledge)	38	26	046	&	70	46	106	F	102	66	146	#102;			
7	7	007	BEL (bell)	39	27	047	'	71	47	107	G	103	67	147	#103;			
8	8	010	BS (backspace)	40	28	050	(72	48	110	H	104	68	150	#104;			
9	9	011	TAB (horizontal tab)	41	29	051)	73	49	111	I	105	69	151	#105;			
10	A	012	LF (NL line feed, new line)	42	2A	052	*	74	4A	112	J	106	6A	152	#106;			
11	B	013	VT (vertical tab)	43	2B	053	+	75	4B	113	K	107	6B	153	#107;			
12	C	014	FF (NP form feed, new page)	44	2C	054	,	76	4C	114	L	108	6C	154	#108;			
13	D	015	CR (carriage return)	45	2D	055	-	77	4D	115	M	109	6D	155	#109;			
14	E	016	SO (shift out)	46	2E	056	.	78	4E	116	N	110	6E	156	#110;			
15	F	017	SI (shift in)	47	2F	057	/	79	4F	117	O	111	6F	157	#111;			
16	10	020	DLE (data link escape)	48	30	060	:	80	50	120	P	112	70	160	#112;			
17	11	021	DC1 (device control 1)	49	31	061	;	81	51	121	Q	113	71	161	#113;			
18	12	022	DC2 (device control 2)	50	32	062	<	82	52	122	R	114	72	162	#114;			
19	13	023	DC3 (device control 3)	51	33	063	=	83	53	123	S	115	73	163	#115;			
20	14	024	DC4 (device control 4)	52	34	064	>	84	54	124	T	116	74	164	#116;			
21	15	025	NAK (negative acknowledge)	53	35	065	?	85	55	125	U	117	75	165	#117;			
22	16	026	SYN (synchronous idle)	54	36	066	@	86	56	126	V	118	76	166	#118;			
23	17	027	ETB (end of trans. block)	55	37	067	A	87	57	127	W	119	77	167	#119;			
24	18	030	CAN (cancel)	56	38	070	B	88	58	130	X	120	78	170	#120;			
25	19	031	EM (end of medium)	57	39	071	C	89	59	131	Y	121	79	171	#121;			
26	1A	032	SUB (substitute)	58	3A	072	D	90	5A	132	Z	122	7A	172	#122;			
27	1B	033	ESC (escape)	59	3B	073	E	91	5B	133	[123	7B	173	#123;			
28	1C	034	FS (file separator)	60	3C	074	F	92	5C	134	\	124	7C	174	#124;			
29	1D	035	GS (group separator)	61	3D	075	G	93	5D	135]	125	7D	175	#125;			
30	1E	036	RS (record separator)	62	3E	076	H	94	5E	136	^	126	7E	176	#126;			
31	1F	037	US (unit separator)	63	3F	077	I	95	5F	137	_	127	7F	177	#127;			

Source: www.LookupTables.com

2.7 Komponen Penyusun Mini Segway

2.7.1 IMU MPU6050

Penggunaan sensor IMU dipilih seri MPU6050 karena mengacu pada beberapa penelitian sebelumnya tentang *segway* yang menggunakan sensor tersebut [4]. MPU6050 memiliki 3-axis *gyroscope* dan 3-axis *accelerometer*. Namun dalam tugas akhir ini hanya menggunakan 1-axis *gyroscope* saja, yaitu axis-Y sebagai sudut *pitch*.



Gambar 2.14 IMU MPU6050

MPU6050 memiliki fitur sebagai berikut :

2.7.1.1 Fitur Gyroscope

Tiga sumbu MEMS *gyroscope* pada MPU-6050 mencakup berbagai fitur :

- Output digital X-, Y- dan Z-axis sensor dengan rentang penuh yang dapat diprogram pengguna dari ± 250 , ± 500 , ± 1000 dan ± 2000 °/sec.
- Sinyal sinkronisasi eksternal yang terhubung ke pin FSYNC mendukung sinkronisasi gambar, video dan GPS.
- Integrasi 16-bit ADC memungkinkan pengambilan sampel gyro secara simultan.
- Peningkatan kestabilan bias dan sensitivitas suhu akan mengurangi kebutuhan akan kalibrasi pengguna.
- Frekuensi rendah akan meningkatkan performa *noise*.
- Besar arus kerja *gyroscope* : 3.6 mA.
- *Standby current* : 5 uA.
- Faktor skala kepekaan kalibrasi pabrik.
- *User self-test*.

2.7.1.2 Fitur Accelerometer

Tiga sumbu MEMS *accelerometer* pada MPU-6050 mencakup berbagai fitur :

- Output digital *triple-axis* dengan rentang skala penuh yang dapat diprogram dari $\pm 2g$, $\pm 4g$, $\pm 8g$ dan $\pm 16g$.
- Integrasi ADC 16-bit memungkinkan pengambilan sampel *accelerometer* secara simultan dan tidak memerlukan *multiplexer* eksternal.
- Besar arus kerja *accelerometer* : 500 uA.
- Besar arus dalam mode *low power* : 10 uA pada 1.25 Hz, 20 uA pada 5 Hz, 60 uA pada 20 Hz, 110 uA pada 40 Hz.
- *Orientation detection and signaling*.
- *Tap detection*.

- *User-programmable interrupts.*
- *High-G interrupt.*
- *User self-test.*

2.7.1.3 Fitur Tambahan

MPU-6050 memiliki beberapa fitur tambahan :

- *9-Axis MotionFusion* oleh *Digital Motion Processor (DMP)* on-chip.
- *Auxiliary master I2C bus* untuk membaca data dari sensor eksternal (misal magnetometer).
- Arus kerja 3.9 mA ketika semua 6 axis motion sensing dan DMP diaktifkan.
- Tegangan *supply* antara 2.375V – 3.46V.
- Fleksibel tegangan referensi VLOGIC mendukung beberapa tegangan antarmuka I2C.
- Paket QFN terkecil dan tertipis untuk perangkat portabel : 4x4x0.9 mm.
- Sensitivitas cross-axis minimal antara *accelerometer* dan sumbu *gyroscope*.
- 1024 byte FIFO buffer mengurangi konsumsi daya dengan membiarkan *host processor* membaca data dan kemudian beralih ke mode daya rendah saat MPU mengumpulkan lebih banyak data.
- Sensor suhu dengan *output* digital.
- Filter digital yang dapat diprogram pengguna untuk *gyroscope*, *accelerometer*, dan sensor temperatur.
- 10.000g *shock tolerant*.
- 400 KHz *Fast Mode I2C* untuk komunikasi data dengan semua register.
- 1 MHz *SPI serial interface* untuk komunikasi data dengan semua register.
- 20 MHz *SPI serial interface* untuk membaca sensor dan register *interrupt*.
- Struktur MEMS ditutup rapat dan teikat pada tingkat wafer.

- *RoHS and Green compliant.*

2.7.1.4 Motion Processing

- Mesin *Internal Digital Motion Processing™* (DMP™) mendukung *3D MotionProcessing* dan algoritma pengenalan isyarat.
- MPU-6050 mengumpulkan data *gyroscope* dan *accelerometer* bersamaan dengan menyinkronkan pengambilan sampel data dengan kecepatan yang ditentukan pengguna. Total data lengkap yang diperoleh MPU-6050 mencakup data *gyroscope* 3-axis, data *accelerometer* 3-axis, dan data suhu. Keluaran MPU yang dihitung ke sistem prosesor juga dapat mencakup data data magnetometer 3-axis.
- FIFO buffer mengurangi persyaratan waktu pada prosesor sistem dengan membiarkan prosesor membaca data FIFO. Kemudian sistem dapat menghemat daya dengan memasukkan *sleep mode* sementara MPU mengumpulkan lebih banyak data.
- *Programmable interrupt* mendukung fitur seperti *gesture recognition*, *panning*, *zooming*, *scrolling*, *tap detection*, dan *shake detection*.
- Filter Low Pass yang dirogram secara digital.
- Fungsi pedometer berkekuatan rendah memungkinkan *processor* untuk *sleep* sementara DMP mempertahankan jumlah langkahnya.

2.7.1.5 Clocking

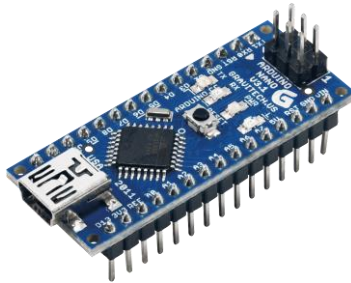
- Generator timing on-chip $\pm 1\%$ pada rentang suhu penuh variasi frekuensi.
- Opsional masukan *external clock* 32.768 KHz atau 19.2 MHz.

2.7.2 Arduino Nano

Pemilihan *microcontroller* Arduino Nano dilakukan karena desain *hardware* hanya menyediakan sedikit ruang kontrol. Sehingga dipilih *microcontroller* Arduino Nano yang memiliki ukuran yang relatif kecil dan cukup jika diletakkan pada desain *hardware*. Arduino Nano digunakan untuk menerima data dari sensor IMU melalui pin I2C. Data sudut yang diterima dari sensor IMU di *filter* sehingga data dari sensor IMU dapat diolah ditahap selanjutnya. Komunikasi data sudut yang diterima dari sensor IMU oleh Arduino Nano selanjutnya dikirim ke *microcontroller* Maple Mini melalui komunikasi serial.

Dalam perancangan ini membutuhkan 2 buah *microcontroller* Arduino Nano karena sensor IMU memberikan 2 buah data sudut yang berbeda disetiap sisinya, dan merupakan 2 data yang tidak saling mempengaruhi satu sama lain. Arduino Nano memiliki spesifikasi sebagai berikut :

- MCU : Atmega328P
- Operating Voltage : 5 V
- Input Voltage (recommended) : 7 V – 12 V
- Digital I/O pins : 14 (of which 6 provide PWM output)
- Analog input pins : 8
- DC Current per I/O Pin : 40mA
- Flash Memory : 32KB (Atmega328) of which 2 KB used by bootloader
- SRAM : 2KB (Atmega328)
- Clock Speed : 16 Mhz
- Dimensions : 45 x 18 x 19mm
- Weight : 6g



Gambar 2.15 *Microcontroller Arduino Nano*

2.7.3 Maple Mini

Microcontroller Maple Mini berfungsi sebagai *main program*, dimana memiliki program *PID controller*. Data sudut yang dikirim oleh Arduino Nano secara serial diterima oleh Maple Mini dan dari *PID controller* menghasilkan *output* berupa PWM. Output PWM ini bertindak sebagai *input* dari *controller* BLDC yang pada tahap selanjut akan menggerakkan motor.

Dalam perancangan ini membutuhkan 2 *microcontroller* Arduino Nano dan Maple Mini agar tidak membani *microcontroller* dengan dua fungsi program, satu untuk mengolah data sensor dari IMU dan satu untuk pengolah data dengan *PID controller* dan menghasilkan output berupa PWM dengan 2 *input* dan 2 *output* yang berbeda dimana tidak saling mempengaruhi. Maple Mini merupakan mikrokontroler berbasis ARM 32-bit yang diproduksi oleh *Leaf labs* dan dapat diprogram dengan mengikuti *style* seperti pada Arduino-IDE. Maple Mini memiliki spesifikasi sebagai berikut :

- MCU : STM32F103RCBT6, 32-bit ARM Cortex M3 microprocessor.
- Clock Speed : 72 Mhz
- 128 KB Flash and 20 KB SRAM
- 34 pin input digital (GPIO)
- 12 pin PWM, 16-bit
- 9 pin input analog (ADC), 12-bit
- 2 SPI peripherals

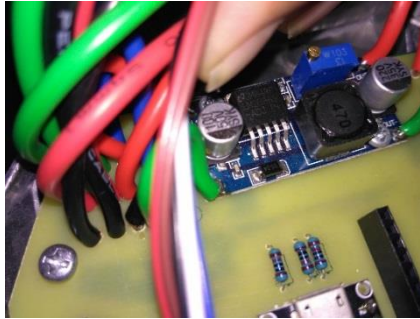


Gambar 2.16 *Microcontroller Maple Mini*

- 2 I2C peripherals
- 7 Channel DMA (Direct Memory Access)
- 3 USART (serial port) peripherals
- 1 advanced and 3 general-purpose timers
- Dedicated USB port of programming and communication
- JTAG
- Nested Vectored Interrupt Controller (NVIC) (including external interrupt on GPIOs)
- Supplies up to 500 mA at 3.3 V, with separate 250 mA digital and analog regulators for low-noise analog performance
- Open source, four layer design
- Support for low power, sleep, and standby modes (<500 uA)
- Operating Voltage : 3.3 V
- Input Voltage (recommended) : 3 V – 12 V
- Dimensions : 2.02” x 0.72”

2.7.4 Buck Converter

Modul *Buck converter* digunakan untuk menurunkan tegangan dari baterai sebesar 36V menjadi 12V dengan arus maksimal 3A sebagai *input power supply* bagi *microcontroller* baik Arduino Nano maupun Maple Mini.



Gambar 2.17 Modul *Buck Converter*

2.7.5 Controller BLDC 350W Full Fitur

Controller ini berfungsi sebagai *driver*/penggerak motor *brushless* DC. Bisa digunakan untuk motor BLDC sepeda listrik, tricycle listrik, sepeda motor listrik dll. *Controller* ini memiliki spesifikasi sebagai berikut :

Power	=	350W – 400W
Tegangan	=	36V – 48V Auto Voltage
Under	=	31,6V untuk 36V. 41V untuk
Voltage		48V
Jumlah FET	=	6 MOSFET
Amper	=	17A
Input throttle	=	1,2V – 4,2V
Derajat motor	=	120 derajat

Dan memiliki beberapa fitur seperti :

- Switch maju – mundur
- Brake signal
- Pedal Assist Sensor untuk sepeda listrik
- Signal speedometer
- Cruise control
- Auto self study (normal forward dan normal backward)



Gambar 2.18 *Controller BLDC 350W Full Fitur*

Keterangan soket :

1. Soket isi 3 besar (Soket power)
 Kabel merah besar terhubung ke + baterai.
 Kabel hitam besar terhubung ke negatif.
 Kabel orange kecil ke kunci kontak lalu ke + baterai.
2. Kabel 3 *phase* motor : kuning, hijau, biru, (kabel besar skun lonjong)
 Ketiga kabel masing – masing terhubung ke motor.
 Dihubungkan dengan 5 kabel hall yang dari motor.
3. Kabel hall : (*socket* 6 isi 5)
 Merah (+5V), hitam (-), kuning, biru, hijau.
 Dihubungkan dengan 5 kabel hall yang dari motor.
4. *Throttle/handle* gas : *socket* isi 3
 Merah (+5V), hitam (-), hijau (*signal data*).
 *pemasangan yang terbalik + dan – menyebabkan *controller* mematikan diri.
5. Kabel putih sepasang = kalibrasi arah putaran
6. *Socket* isi 2 hitam dan biru = *cruise control*
7. *Socket* isi 2 hitam dan putih = *switch* rem
8. *Socket* isi 2 hitam dan abu – abu = *switch* maju – mundur
9. Coklat tunggal = *signal* speedometer
10. Ungu tunggal = *signal* hi level brake
11. Tiga kabel pin kecil = Pedal Assist Sensor

2.7.6 BLDC Motor

Mini segway membutuhkan gerakan motor yang detail dan stabil, karena jika tidak akan mempengaruhi percepatan sudut yang mengakibatkan respon menjadi lamban dan pengendara akan terjatuh. Oleh karena itu dalam perancangan kali ini menggunakan motor jenis *brushless* karena memiliki torsi per berat badan lebih besar, kecepatan motor yang stabil dan lebih efisien, suara putaran lebih halus, dan tidak membutuhkan *gear* tambahan bila dibandingkan dengan motor jenis *brushed*. Namun untuk menggerakkannya dibutuhkan ESC (*Electronic Speed Control*) / *Electric Controller*. Walaupun merupakan motor listrik sinkron AC 3 fasa, motor ini tetap disebut dengan BLDC karena pada implementasinya BLDC menggunakan sumber DC sebagai sumber energi utama yang kemudian diubah menjadi tegangan AC dengan menggunakan *inverter* 3 fasa.

Tujuan dari pemberian tegangan AC 3 fasa pada stator BLDC adalah menciptakan medan magnet putar stator untuk menarik magnet rotor. Oleh karena tidak adanya *brush* pada motor BLDC, untuk menentukan *timing* komutasi yang tepat pada motor sehingga didapatkan torsi dan kecepatan yang konstan, diperlukan 3 buah *hall sensor*. Pada *hall sensor*, *timing* komutasi ditentukan dengan cara mendeteksi medan magnet rotor dengan menggunakan 3 buah *hall sensor* untuk mendapatkan 6 kombinasi *timing* yang berbeda.



Gambar 2.19 Motor *Brushless*

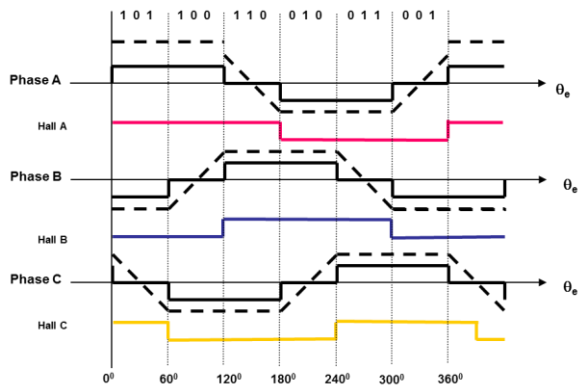
Motor *brushless* yang digunakan memiliki spesifikasi sebagai berikut :

- Motor code : WDF36V1601210065
WDF36V1601230822
- Power : 350W x 2
- Max speed : 15 km/h
- Rated speed : 800 rpm
- Diameter : 6.5 inch (17 cm)

Kabel 3 warna (kuning, biru, hijau) adalah kabel 3 fasa motor. Kabel 5 warna (merah, kuning, biru, hijau, hitam) adalah kabel *hall sensor*.



Gambar 2.20 Kabel Motor *Brushless*



Gambar 2.21 Sinyal 3 Fasa Dengan *Hall Sensor*

2.7.7 Baterai Pack

Power supply mini segway menggunakan baterai *Lithium-Ion Pack* 36V dengan 22 buah baterai. Nama produk Hcpower-10S2P, model 10S2P A01, dan memiliki dimensi 135*90*60mm.

2.7.8 Saklar

Saklar berfungsi sebagai ON/OFF pemutus *power supply* dari baterai. Saklar yang digunakan memiliki 3 pin seperti pada Gambar 3.10. parameter yang digunakan adalah :

- Pin : *Normally Open (NO), Normally Closed (NC)*
- Batas arus :
 - 250 VAC, 6A
 - 125VAC, 10A



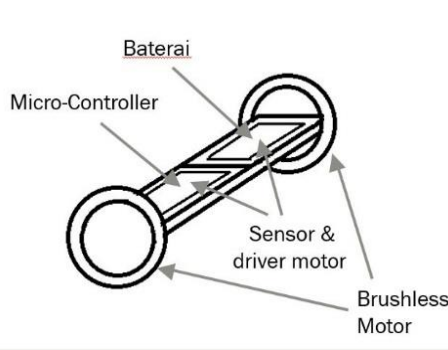
Gambar 2.22 Baterai Pack



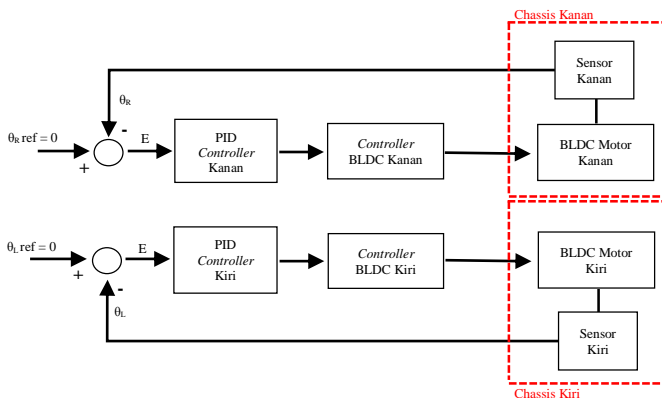
Gambar 2.23 Saklar

BAB III PERANCANGAN SISTEM

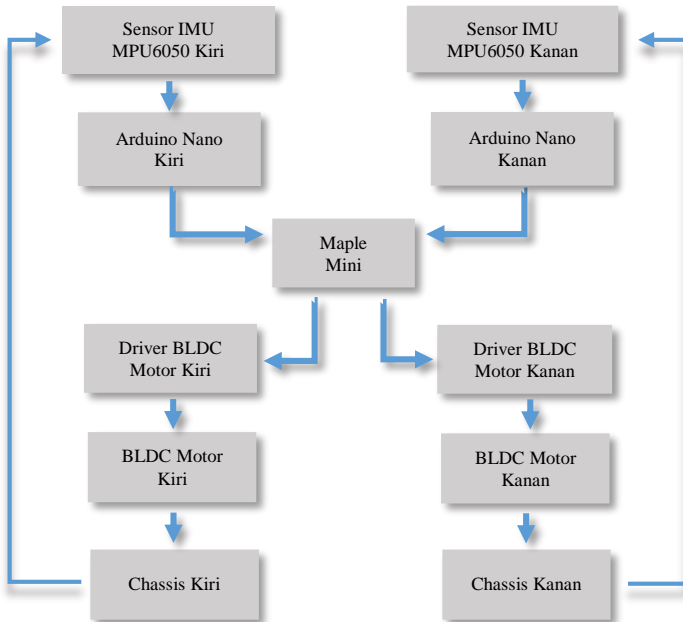
Pada bab ini akan dijelaskan perancangan sistem secara keseluruhan mulai dari perancangan *hardware* yang meliputi perancangan desain *hardware*, penggunaan *microcontroller*, *controller* BLDC sebagai *driver* motor, dan motor BLDC hingga perancangan *software* pemrograman PID.



Gambar 3.1 Desain Keseluruhan Perancangan *Hardware*



Gambar 3.2 Blok Diagram Sistem Kontrol



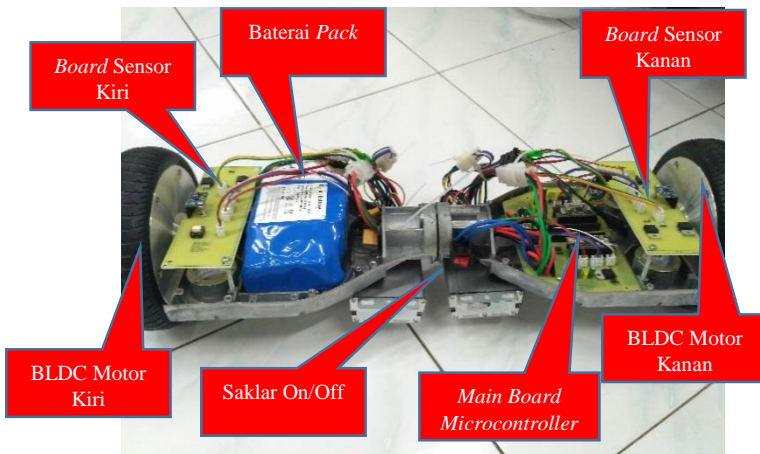
Gambar 3.3 Blok Diagram Sistem

Blok diagram dari sistem kontrol ditunjukkan seperti pada Gambar 3.2. *Input* sistem berupa sudut referensi (θ_{ref}) 0° . Ketika *chassis* dimiringkan maka akan sudut θ akan berubah yang menghasilkan nilai *error* (E). Nilai *error* ini akan menjadi *input* pada proses selanjutnya. Kemudian dengan *PID controller*, *input* dari *error* diolah sehingga menghasilkan keluaran *output* berupa PWM. PWM akan menjadi *input* pada *controller* BLDC untuk menggerakkan BLDC motor. BLDC Motor akan bergerak yang menyebabkan *chassis* berubah pada keadaan seimbang kembali. Ketika sudut θ tidak bernilai 0° (terdapat *error* sudut) maka akan di *feedback* sehingga akan mengupdate sudut *input* untuk *PID controller* dan akan mengeluarkan PWM yang akan menggerakkan motor BLDC sehingga nilai *error* menjadi 0.

3.1 Desain *Hardware Mini Segway*



Gambar 3.4 Desain *Hardware Mini Segway* Tampak Atas



Gambar 3.5 Desain *Hardware Mini Segway* Tampak Bawah

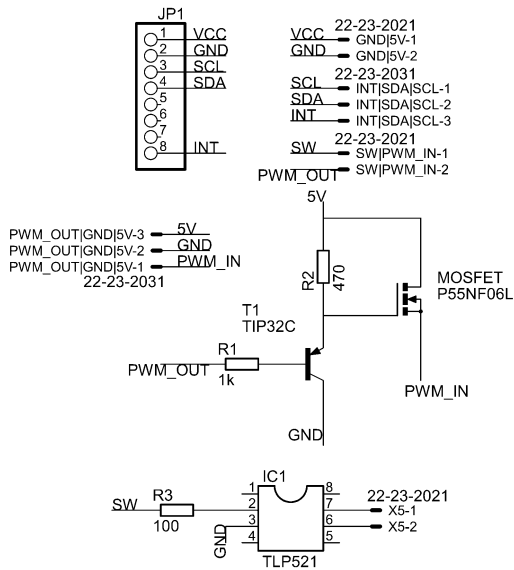
3.1.1 Board Sensor

Terdapat 2 sensor IMU MPU6050 yang masing – masing diletakkan pada kedua sisi pijakan *mini segway*. Pada perancangan ini data dari sensor IMU hanya menggunakan data kemiringan sudut saja. Pada sensor IMU juga hanya diambil 1 Axis yaitu Axis-Y sebagai sudut *pitch*, dan didapatkan sudut dari -90° sampai dengan 90° . Komunikasi untuk mengirim data sudut dari sensor IMU dengan komunikasi I2C. Terdapat optocoupler sebagai *switching* pergerakan *mini segway* maju dan mundur.



Gambar 3.6 Board Sensor

Perancangan skematik maupun perancangan *board* PCB pada rangkaian *board* sensor secara keseluruhan dan komponen yang digunakan sesuai pada Gambar 3.23, Gambar 2.2 dan Tabel 3.3.



Gambar 3.7 Skematik Rangkaian Keseluruhan *Board Sensor*

Tabel 3.1 Tabel Komponen Pada Rangkaian *Board Sensor*

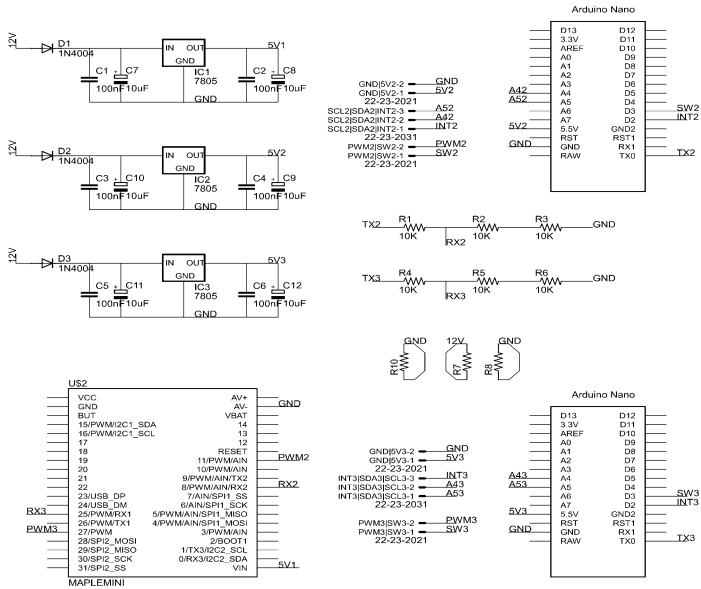
No.	Nama Komponen	Kode	Nilai
1	Resistor	-	100 Ohm, 470 Ohm, 1 KOhm,
2	Transistor	TIP32	-
3	MOSFET	STP55NF06L	-
4	Optocoupler	TLP521	-
5	Connector XH 2-pin	-	-
6	Connector XH 3-pin	-	-

sensor IMU sisi kanan, dan Arduino 2 mengolah data dari sensor IMU sisi kiri. Maple Mini akan mengolah data sudut dari masing – masing Arduino dan menghasilkan *output* PWM sebagai *input controller* BLDC.

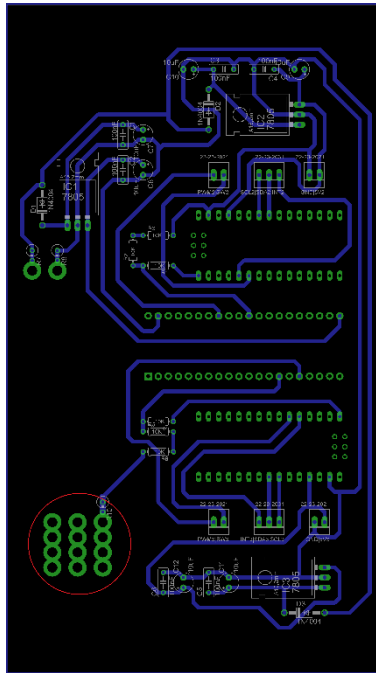
Perancangan skematik dan *board PCB* pada rangkaian *main board* secara keseluruhan, dan tabel denah penetapan kabel sesuai pada Gambar 3.16, Gambar 3.22, dan Tabel 3.2.



Gambar 3.9 *Main Board*



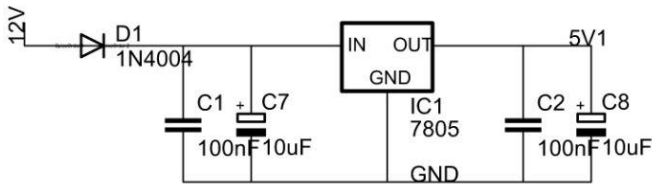
Gambar 3.10 Skematik Rangkaian Keseluruhan *Main Board*



Gambar 3.11 Board PCB Keseluruhan Pada Main Board

Tabel 3.2 Tabel Denah Penetapan Kabel Pada Main Board

	A	B	C
1	<i>Input +36V Buck Converter</i>	<i>Switch saklar</i>	<i>GND Buck Converter</i>
2	<i>Switch saklar</i>	<i>+Baterai</i>	<i>GND baterai</i>
3	<i>Enable controller 2 (kanan)</i>	<i>+Controller 2 (kanan)</i>	<i>GND controller 2 (kanan)</i>
4	<i>Enable controller 1 (kiri)</i>	<i>+Controller 1 (kiri)</i>	<i>GND controller 1 (kiri)</i>



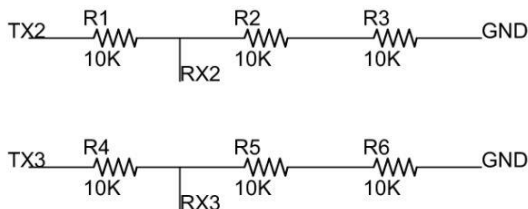
Gambar 3.12 Skematik Rangkaian Penstabil Tegangan

Tabel 3.3 Tabel Komponen Pada Rangkaian Penstabil Tegangan

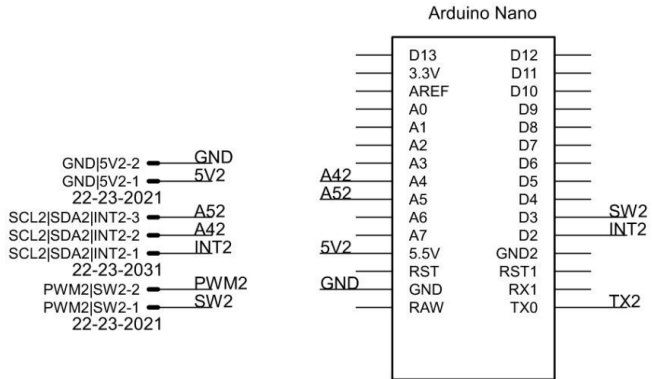
No.	Nama Komponen	Kode	Nilai
1	Dioda penyearah	1N4007	-
2	Regulator	7805	-
3	Kapasitor mylar	-	100nF
4	Kapasitor bipolar	-	10uF

Perancangan skematik beserta komponen yang digunakan pada rangkaian penstabil tegangan sebagai *power supply* untuk *microcontroller* baik Arduino maupun Maple Mini dengan *input* 12V dari *output buck converter* seperti pada Gambar 3.17 dan Tabel 3.1.

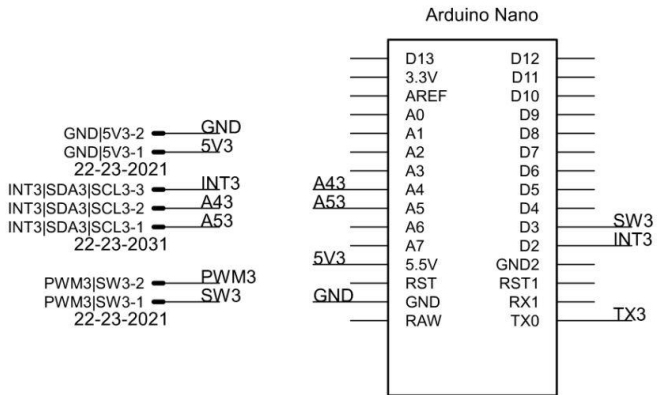
Perancangan skematik rangkaian penurun tegangan dari 5V menjadi 3.3V sebagai komunikasi serial antara Arduino dan Maple Mini, skematik rangkaian Arduino bagian atas maupun bawah, serta skematik rangkaian Maple Mini sesuai pada Gambar 3.21, Gambar ...,



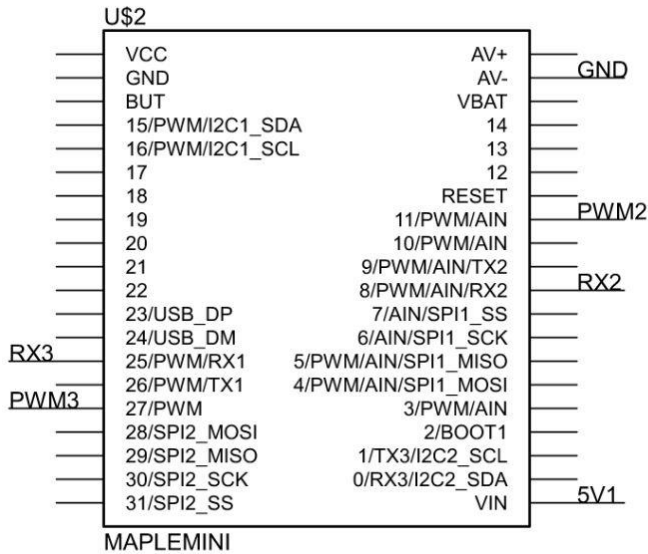
Gambar 3.13 Skematik Rangkaian Komunikasi Serial



Gambar 3.14 Skematik Rangkaian Arduino Bagian Atas



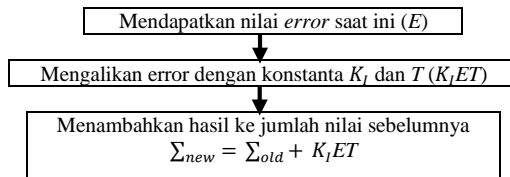
Gambar 3.15 Skematik Rangkaian Arduino Bagian Bawah



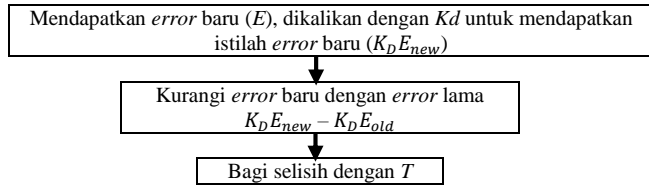
Gambar 3.16 Skematik Rangkaian Maple Mini

3.2 Desain Software Mini Segway

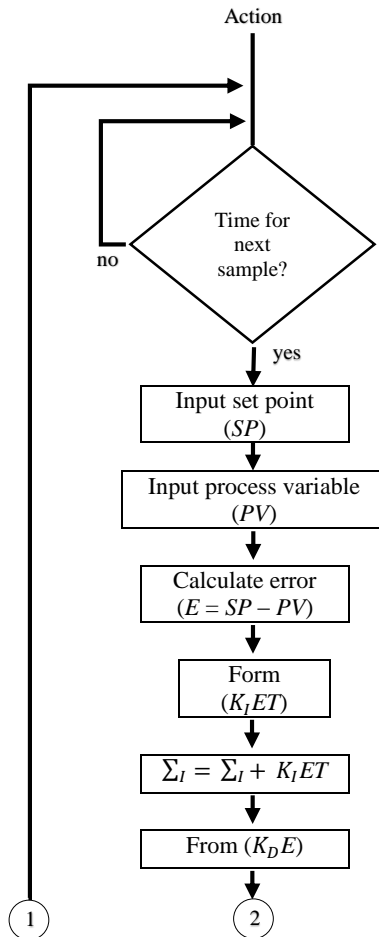
Pemrograman *software* pada Arduino Nano menggunakan aplikasi Arduino-IDE. Sedangkan Pemrograman *software* pada Maple Mini menggunakan aplikasi Maple-IDE. Arduino Nano melakukan kalibrasi sudut sensor, pengolahan pembacaan data sudut dari sensor IMU, dan Maple Mini melakukan pengolahan data sudut yang diberikan oleh Arduino Nano menjadi output PWM yang menggerakkan motor BLDC dengan kontrol PID.

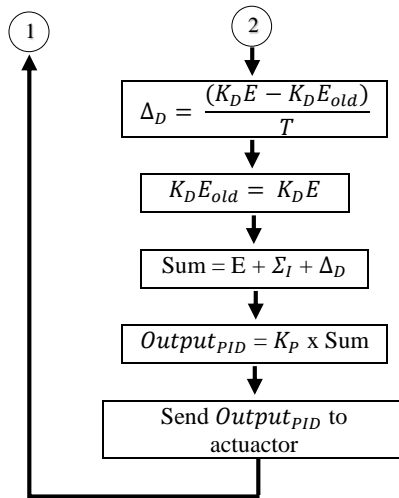


Gambar 3.17 Flowchart Untuk Mendapatkan Integral



Gambar 3.18 *Flowchart* Untuk Mendapatkan Derivatif



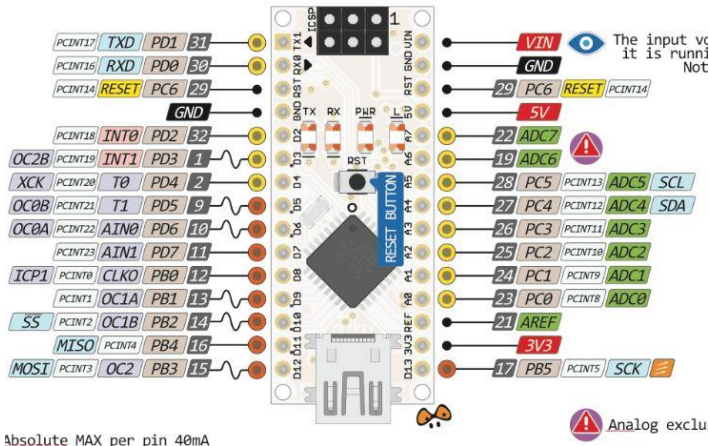


Gambar 3. 19 *Flowchart* Persamaan PID

Algoritma : Penentuan Output PWM dengan Kontrol PID	
1	Function output_PID(Kp, Ki, Kd, SP, PV, E)
Input	Kp, Ki, dan Kd untuk paramater kontrol PID SP untuk parameter set point kontrol PID PV merupakan data sudut sebenarnya E merupakan error sudut
Output	output PID untuk penentuan error posisi keseimbangan
2	E = SP – PV
3	Jumlah_error_integral = Jumlah_error_integral + Ki*E
4	Derivative_kontrol = (Kd*E – Kd*E_lama)
5	Kd*Error_lama = Kd*Error
6	Sum = (E + Jumlah_error_integral + Derivative_kontrol)
7	output_PID = Kp*Sum
8	return (output_PID)

Gambar 3.20 Algoritma Penghitungan Nilai *Output* PID

3.3 Perancangan *Pin Out* Yang Digunakan Pada Arduino



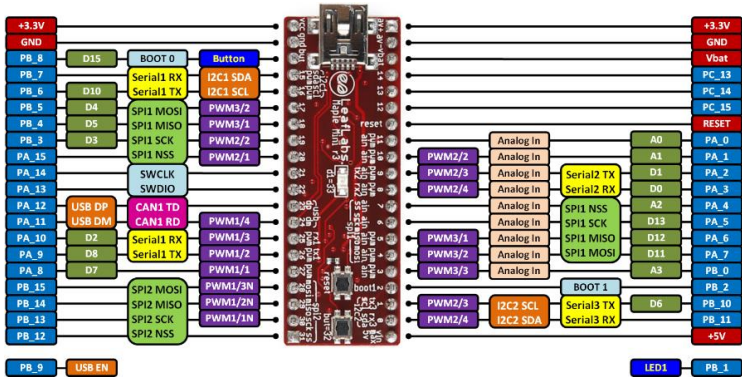
Gambar 3.21 *Pin Out* Arduino Nano

Tabel 3.4 Tabel Konfigurasi *Pin Out* Pada Arduino Nano Dengan MPU6050

MPU6050	Pin Arduino Nano	Keterangan
VCC	5V	Input tegangan +5V
GND	GND	GND
SDA	A4	SDA
SCL	A5	SCL
INT	D2	Interrupt
-	D3	On/Off maju mundur. Dihubungkan dengan optocoupler pada board sensor yang tersambung dengan kabel hitam dan abu – abu pada controller BLDC

-	TX	TX sebagai pengiriman data sudut dengan komunikasi serial ke Maple Mini
---	----	---

3.4 Perancangan *Pin Out* Yang Digunakan Pada Maple Mini



Gambar 3.22 *Pin Out* Maple Mini

Tabel 3.5 Tabel Konfigurasi *Pin Out* Maple Mini Dengan Arduino Nano

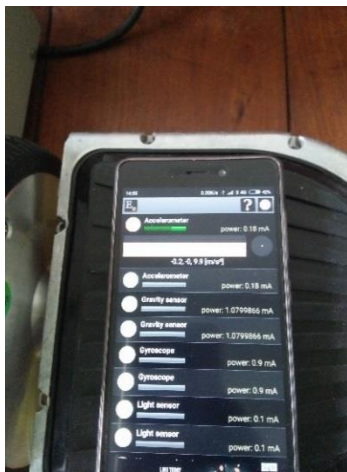
Pin Arduino Nano	Pin Maple Mini	Keterangan
TX	25, 8	Pin 25 sebagai RX1 menerima data dari Arduino 1. Pin 8 sebagai RX2 menerima data dari Arduino 2.
-	Vin	Input tegangan +5V
-	aV-	GND
-	27	Output PWM1
-	11	Output PWM2

BAB IV PENGUJIAN

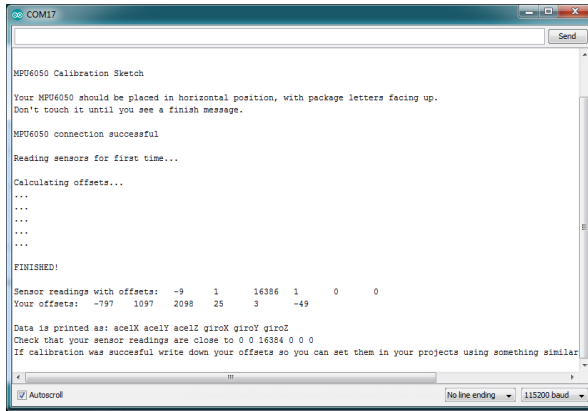
Pada bab ini akan dibahas mengenai pengujian dari sistem yang telah dirancang. Bab ini bertujuan untuk mengetahui apakah tujuan dalam perancangan sistem pada tugas akhir ini telah terlaksana atau tidak. Pengujian pada bab ini terdiri dari pengujian kalibrasi sensor IMU MPU6050, pengujian pengiriman data sudut menggunakan komunikasi serial, serta pengujian pengolahan data sudut menjadi keluaran PWM sebagai penggerak motor.

4.1 Kalibrasi Sensor IMU MPU6050

Pengujian kalibrasi sensor ini dilakukan untuk menentukan posisi awal sensor. Sensor diposisikan seperti pada Gambar 4.1. *Mini segway* diposisikan mendatar pada posisi siap imbangan jika dikendarai dengan *board* sensor berada dibawah pijakan. Kemudian dengan bantuan dari aplikasi Elixir pada *handphone* dilakukan kalibrasi posisi 0° *mini segway* dan akan menghasilkan nilai *offset* yang akan dimasukkan ke program *offset* sudut sebagai posisi imbangan *mini segway*.



Gambar 4.1 Kalibrasi Sensor IMU MPU6050 Dengan Aplikasi Elixir Pada Handphone



Gambar 4.2 Hasil Kalibrasi Sensor IMU MPU6050

Tabel 4.1 Data *Offset* Masing – Masing Sensor

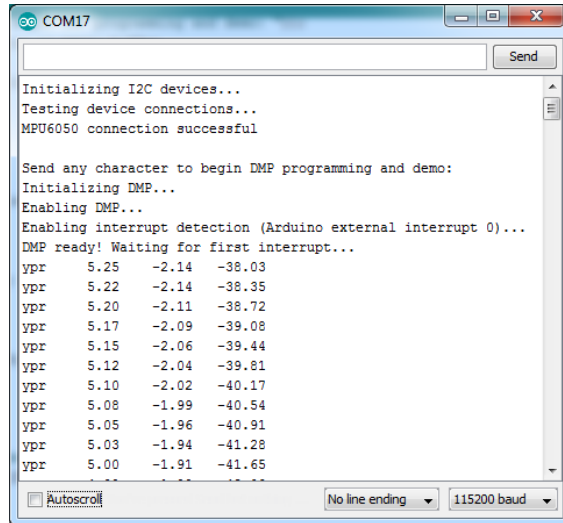
Data <i>offset</i>	Sensor IMU Kanan	Sensor IMU Kiri
<i>acceleroX</i>	1859	-839
<i>acceleroY</i>	956	1087
<i>acceleroZ</i>	2506	2101
<i>gyroX</i>	40	22
<i>gyroY</i>	3	22
<i>gyroZ</i>	54	-53

4.2 Pengujian Sensor IMU MPU6050

Data *offset* sudut yang telah didapatkan sebelumnya dimasukkan kedalam *offset* program :

```
// supply your own gyro offsets here, scaled for min sensitivity
mpu.setXGyroOffset(40);
mpu.setYGyroOffset(3);
mpu.setZGyroOffset(54);
mpu.setZAccelOffset(2506); // 1688 factory default for my test chip
```

Kemudian dilakukan pengujian untuk menentukan apakah sensor sudah berfungsi dengan benar dan data sudut yang didapatkan mendekati nilai yang sebenarnya atau belum.



Gambar 4.3 Data Hasil Pengujian Sensor IMU MPU6050

Data sudut yang diperoleh sudah menunjukkan nilai yang benar. Kemudian untuk *input* sudut *mini segway* hanya digunakan nilai sudut *pitch* (data kedua dari Gambar 4.3).

4.3 Pengiriman Data Sudut

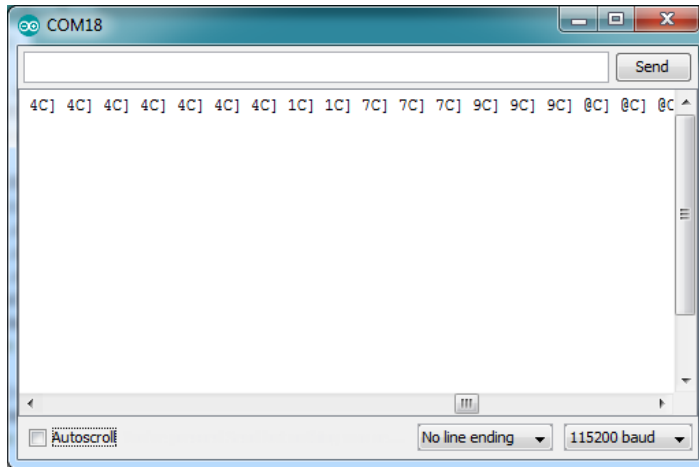
Pengiriman data sudut dari Arduino Nano dilakukan dengan mengirimkan data sudut per byte melalui komunikasi serial menuju *microcontroller* Maple Mini. Komunikasi serial menggunakan Baud Rate 9600 baik pada Arduino Nano maupun pada Maple Mini. Pengiriman data serial hanya satu arah melalui TX saja. Jadi Arduino Nano hanya mengirim data, tidak ada data balik yang diterima.

```

a[0]='';
if(dataYaw<0) {a[1]=1;a[2]=(uint16_t) (dataYaw*-1000)/256;a[3]=(uint16_t) (dataYaw*-1000)/256;}
else {a[1]=0;a[2]=(uint16_t) (dataYaw*1000)/256;a[3]=(uint16_t) (dataYaw*1000)/256;}

Serial.write(a[0]);
Serial.write(a[1]);
Serial.write(a[2]);
Serial.write(a[3]);
delay(100);
  
```

Gambar 4.4 Program Pengiriman Data Sudut

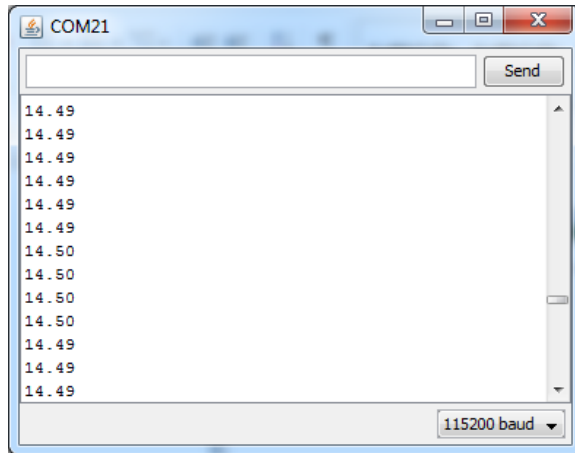


Gambar 4.5 Pengiriman Data Sudut Per Byte

Data `a[0]` sebagai *header* dimana untuk penanda awalan data. Data `a[1]` adalah data posisi sudut + atau – dengan menggantinya menjadi 1 atau 0. Data `a[2]` adalah data 2 digit sudut *pitch* didepan koma. Dan data `a[3]` adalah data 2 digit sudut dibelakang koma.

4.4 Penerimaan Data Sudut

Pengujian dilakukan pada Maple Mini untuk menerima data sudut yang dikirim dari Arduino Nano dengan komunikasi serial. Maple Mini hanya menerima data sudut dan tidak mengirim data apapun kepada Arduino Nano. Dihasilkan data sudut yang diterima sama dengan data yang dikirimkan.



Gambar 4.6 Penerimaan Data Sudut

4.5 Pengolahan Data Sudut

Pengujian dilakukan terhadap program PID pada Maple-IDE. Diawali *tunning* PID secara manual dengan memberikan nilai $K_p = 1$, $K_i = 0$, $K_d = 0$ dan menaikkan nilai K_p sehingga respon sistem mencapai *set point* yang diinginkan. *Error* PID dalam program didapatkan dari nilai sudut yang terbaca oleh sensor yang dikurangi dengan setpoint yang bernilai 0.

Dengan karakteristik dari gerak *chassis mini segway* yang memiliki batas maksimal, pengujian dilakukan untuk menentukan batas minimal nilai K_p yang dibutuhkan sehingga *chassis* dapat melakukan pergerakan rotasi maksimal (*chassis* kanan miring ke depan secara penuh dan *chassis* kiri miring ke belakang secara penuh, atau sebaliknya) dengan beban 1.5 Kg. Pengujian pertama dilakukan dengan nilai $K_p = 1$



Gambar 4.7 Pengujian Penentuan Batas Minimal Nilai Kp

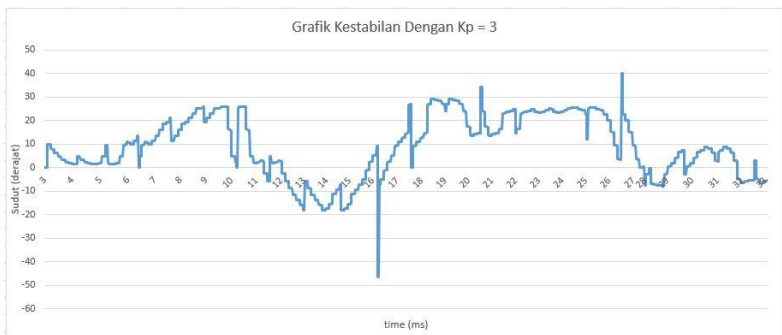


Gambar 4.8 Grafik Respon Sudut Dengan Nilai Kp = 1

Hasil pengujian pertama dengan nilai $K_p = 1$ didapatkan bahwa motor belum bergerak pada posisi sesuai Gambar 4.7 dengan hasil grafik respon sudut terlihat pada gambar 4.8. Karena motor belum bergerak, maka pengujian kedua dilakukan dengan menaikkan nilai K_p menjadi 2.



Gambar 4.9 Grafik Respon Sudut Dengan Nilai $K_p = 2$



Gambar 4.10 Grafik Respon Sudut Dengan Nilai $K_p = 3$

Hasil pengujian kedua dengan nilai $K_p = 2$ didapatkan bahwa motor masih belum bergerak pada posisi sesuai Gambar 4.7 dengan hasil grafik respon sudut terlihat pada gambar 4.9. Karena motor belum bergerak, maka pengujian ketiga dilakukan dengan menaikkan nilai K_p menjadi 3.

Hasil pengujian ketiga dengan nilai $K_p = 3$ didapatkan bahwa motor masih belum bergerak pada posisi sesuai Gambar 4.7 dengan hasil grafik respon sudut terlihat pada gambar 4.10. Karena motor belum bergerak, maka pengujian keempat dilakukan dengan menaikkan nilai K_p menjadi 4.



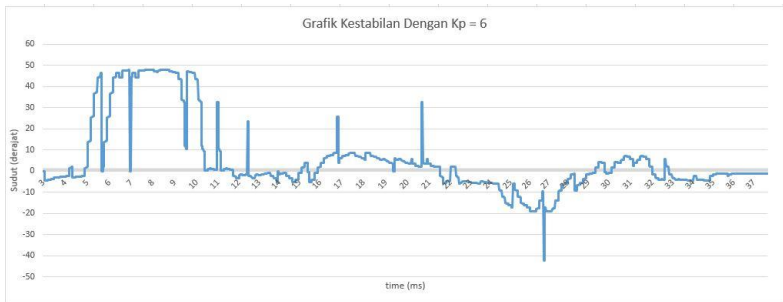
Gambar 4.11 Grafik Respon Sudut Dengan Nilai $K_p = 4$



Gambar 4.12 Grafik Respon Sudut Dengan Nilai $K_p = 5$

Hasil pengujian keempat dengan nilai $K_p = 4$ didapatkan bahwa motor masih belum bergerak pada posisi sesuai Gambar 4.7 dengan hasil grafik respon sudut terlihat pada gambar 4.11. Karena motor belum bergerak, maka pengujian kelima dilakukan dengan menaikkan nilai K_p menjadi 5.

Hasil pengujian kelima dengan nilai $K_p = 5$ didapatkan bahwa motor masih belum bergerak pada posisi sesuai Gambar 4.7 dengan hasil grafik respon sudut terlihat pada gambar 4.12. Karena motor belum bergerak, maka pengujian keenam dilakukan dengan menaikkan nilai K_p menjadi 6.



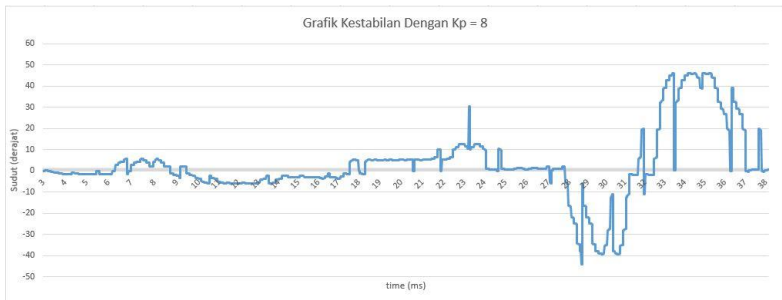
Gambar 4.13 Grafik Respon Sudut Dengan Nilai $K_p = 6$



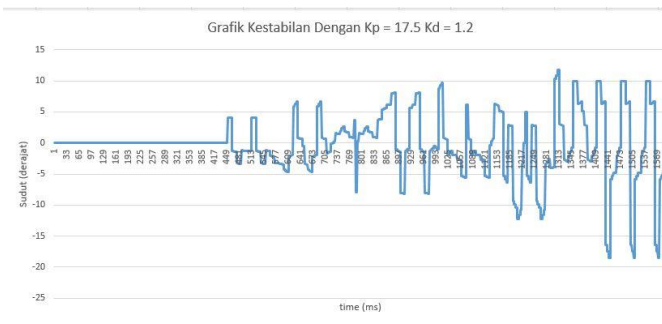
Gambar 4.14 Grafik Respon Sudut Dengan Nilai $K_p = 7$

Hasil pengujian keenam dengan nilai $K_p = 6$ didapatkan bahwa motor masih belum bergerak pada posisi sesuai Gambar 4.7 dengan hasil grafik respon sudut terlihat pada gambar 4.13. Karena motor belum bergerak, maka pengujian ketujuh dilakukan dengan menaikkan nilai K_p menjadi 7.

Hasil pengujian ketujuh dengan nilai $K_p = 7$ didapatkan bahwa motor masih belum bergerak pada posisi sesuai Gambar 4.7 dengan hasil grafik respon sudut terlihat pada gambar 4.14. Karena motor belum bergerak, maka pengujian kedelapan dilakukan dengan menaikkan nilai K_p menjadi 8.



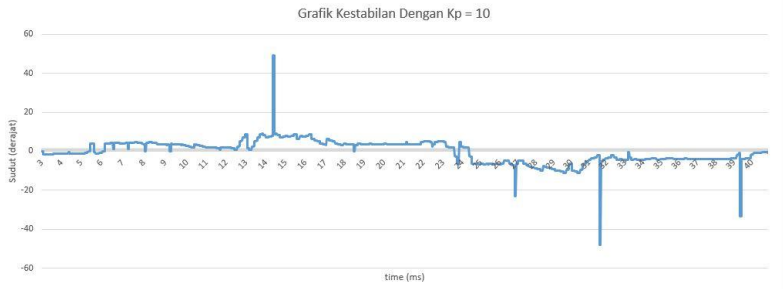
Gambar 4.15 Grafik Respon Sudut Dengan Nilai $K_p = 8$



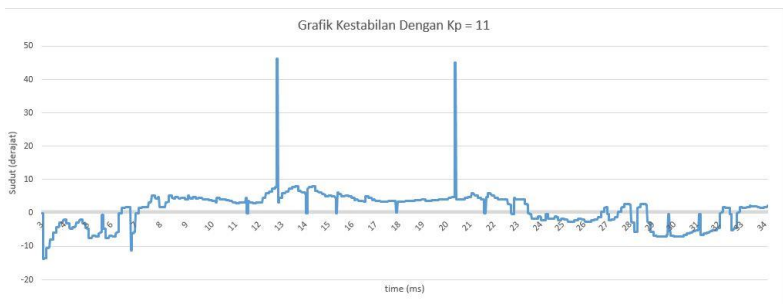
Gambar 4.16 Grafik Respon Sudut Dengan Nilai $K_p = 9$

Hasil pengujian kedelapan dengan nilai $K_p = 8$ didapatkan bahwa motor masih belum bergerak pada posisi sesuai Gambar 4.7 dengan hasil grafik respon sudut terlihat pada gambar 4.15. Karena motor belum bergerak, maka pengujian kesembilan dilakukan dengan menaikkan nilai K_p menjadi 9.

Hasil pengujian kesembilan dengan nilai $K_p = 9$ didapatkan bahwa motor sudah merespon dengan melakukan pergerakan yang pelan dengan hasil grafik respon sudut terlihat pada gambar 4.16. Karena motor berputar lemah dengan posisi sesuai Gambar 4.7, maka pengujian kesepuluh dilakukan dengan menaikkan nilai K_p menjadi 10.



Gambar 4.17 Grafik Respon Sudut Dengan Nilai Kp = 10



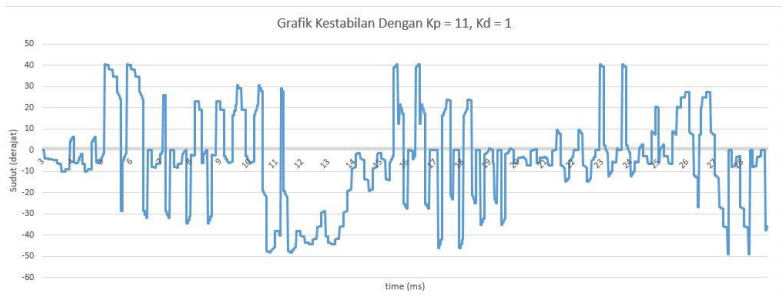
Gambar 4.18 Grafik Respon Sudut Dengan Nilai Kp = 11

Hasil pengujian kesepuluh dengan nilai $K_p = 10$ sesuai grafik respon sudut yang terlihat pada Gambar 4.17 didapatkan bahwa motor berputar sesuai pada posisi sesuai Gambar 4.7, namun belum bisa melakukan pergerakan dengan beban yang ada. Sehingga pada pengujian kesebelas dilakukan pengujian dengan nilai K_p sebesar 11.

Hasil pengujian kesebelas dengan nilai $K_p = 11$ didapatkan bahwa *mini segway* dengan posisi seperti Gambar 4.7 dapat bergerak dan mengangkat beban, terlihat juga respon sudut pada Gambar 4.18, namun respon terlalu cepat. Sehingga pada pengujian ke-12 dilakukan penambahan kontrol derivatif sebesar 1 untuk mengurangi respon yang berlebihan.

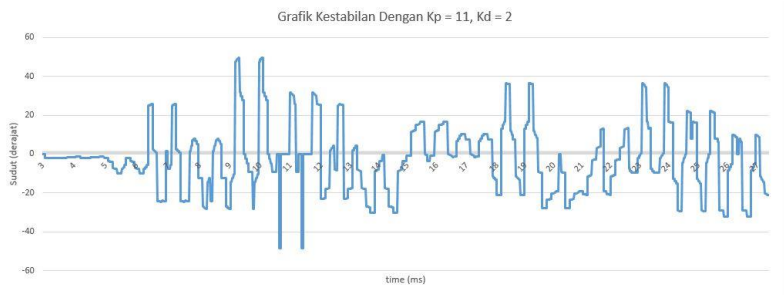


Gambar 4.19 Pengujian Keseimbangan *Mini Segway*



Gambar 4.20 Grafik Respon Sudut Dengan Nilai $K_p = 11$ dan $K_d = 1$

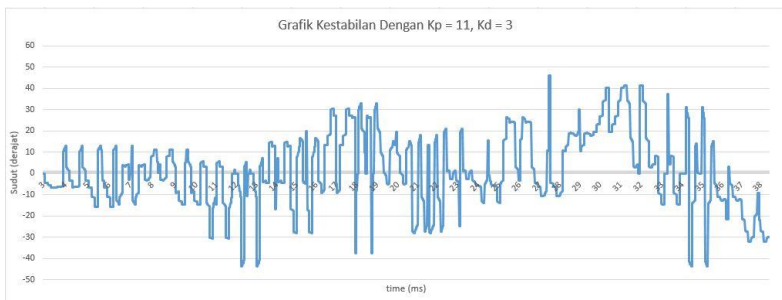
Hasil pengujian ke-12 dengan nilai $K_p = 11$ dan $K_d = 1$ didapatkan bahwa *mini segway* sudah melakukan pergerakan osilasi pada posisi *chassis* 0° sesuai grafik respon sudut yang terlihat pada Gambar 4.20. Karena masih belum stabil maka pada pengujian ke-13 dilakukan pengujian dengan menambah nilai K_d menjadi 2.



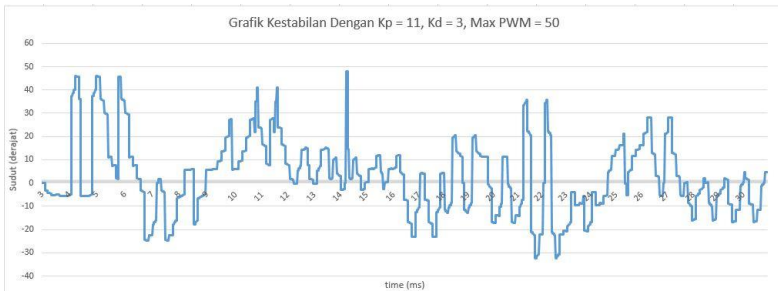
Gambar 4.21 Grafik Respon Sudut Dengan Nilai $K_p = 11$ dan $K_d = 2$

Hasil pengujian ke-13 dengan nilai $K_p = 11$ dan $K_d = 2$ didapatkan bahwa *mini segway* masih melakukan pergerakan osilasi pada posisi *chassis* 0° sesuai grafik respon sudut yang terlihat pada Gambar 4.21. Karena masih belum stabil maka pada pengujian ke-14 dilakukan pengujian dengan menambah nilai K_d menjadi 3.

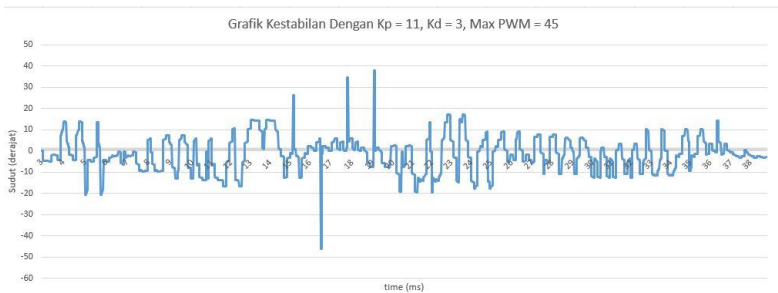
Hasil pengujian ke-14 dengan nilai $K_p = 11$ dan $K_d = 3$ didapatkan bahwa *mini segway* masih melakukan pergerakan osilasi pada posisi *chassis* 0° dengan kecepatan gerak motor yang cepat sesuai grafik respon sudut yang terlihat pada Gambar 4.22. Pengujian selanjutnya dilakukan dengan membatasi kecepatan maksimal pada motor dengan mengatur keluaran PWM. Pengujian ke-15 diberikan batas maksimal keluaran PWM sebesar 50.



Gambar 4.22 Grafik Respon Sudut Dengan Nilai $K_p = 11$ dan $K_d = 3$



Gambar 4.23 Grafik Respon Sudut Dengan Nilai $K_p = 11$, $K_d = 3$, dan Max PWM = 50



Gambar 4.24 Grafik Respon Sudut Dengan Nilai $K_p = 11$, $K_d = 3$, dan Max PWM = 45

Hasil pengujian ke-15 dengan nilai $K_p = 11$, $K_d = 3$ dan PWM maksimal sebesar 50 didapatkan bahwa *mini segway* masih melakukan pergerakan osilasi pada posisi *chassis* 0° dengan kecepatan gerak motor yang belum stabil sesuai grafik respon sudut yang terlihat pada Gambar 4.23. Pengujian selanjutnya dilakukan untuk memperlambat kecepatan motor dengan mengurangi nilai maksimal keluaran PWM. Pengujian ke-16 diberikan batas maksimal keluaran PWM sebesar 45.

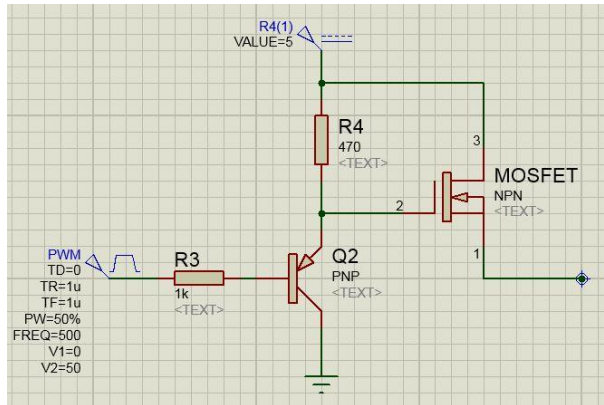
Hasil pengujian ke-16 dengan nilai $K_p = 11$, $K_d = 3$ dan PWM maksimal sebesar 45 untuk *chassis mini segway* bagian kanan, dan dengan cara yang sama didapatkan nilai PWM maksimal untuk *chassis mini segway* bagian kiri sebesar 70, didapatkan bahwa *mini segway* mampu melakukan pergerakan dengan halus sesuai perintah

pengguna, baik melakukan pergerakan maju, mundur, belok kanan, belok kiri, rotasi ke kanan dan rotasi ke kiri seperti pada Gambar 4.24, namun *mini segway* tidak bisa melakukan keseimbangan pada posisi 0° terhadap *chassis*. Hal ini dikarenakan adanya *death zone* yang terlalu besar yaitu antara -3.18° (posisi *chassis* maju ke belakang) sampai dengan 3.43° (posisi *chassis* maju ke depan).

4.6 Pengujian Controller BLDC

Pengujian dilakukan terhadap *controller* BLDC sebagai *driver*/penggerak motor BLDC. Untuk melihat respon bagaimana *controller* BLDC dapat bekerja, pertama dilakukan dengan memasukkan sinyal PWM langsung dari Maple Mini ke *connector controller* BLDC. Dengan mengaktifkan *counter up* pada PWM dari 0 sampai 255, dilihat bagaimana respon gerak motor. Dari pengujian dihasilkan respon motor tidak bergerak sama sekali. Hal ini dikarenakan *output* tegangan PWM dari Maple Mini sebesar 3.3V dan dengan besar tegangan ini tidak bisa dijadikan *input controller* BLDC. Sehingga digunakan rangkaian penguat tegangan dengan menggunakan 1 MOSFET dan 1 Transistor. Dengan rangkaian seperti pada Gambar 4.25, *ouput* tegangan PWM pada MOSFET menjadi 4.3V dan dapat menggerakkan motor.

Controller BLDC memiliki karakteristik yaitu dapat bekerja jika diberikan *input* tegangan minimal 1.2V dan *input* tegangan maksimal sebesar 4.2V. Sehingga ketika tegangan *input* kurang dari 1.2V motor tidak akan bekerja. PWM minimal yang digunakan adalah 75 atau Duty-Cycle sebesar 29.4%. Dalam pengujian yang dilakukan pada *controller* BLDC didapatkan bahwa terdapat *death zone* yang terlalu besar, yaitu daerah mati ketika diberikan *input* tegangan yang besarnya kurang dari +1.2V akan mengakibatkan *input* diabaikan (akan dianggap *input* sama dengan 0), sehingga motor tidak mengalami pergerakan. Karakteristik dari *controller* BLDC lainnya adalah dapat bekerja apabila diberikan tegangan rata – rata (PWM) sebesar 0 terlebih dahulu dan akan bekerja kemudian ketika diberikan *input* tegangan lebih besar dari +1.2V. Dari pengujian sistem didapatkan besarnya nilai *death zone* pada *mini segway* sekitar 6.65° yaitu antara -3.18° (posisi *chassis* maju ke belakang) sampai dengan 3.43° (posisi *chassis* maju ke depan), dan respon pergantian arah gerak maju-mundur dari *controller* BLDC terlalu lambat.



Gambar 4.25 Rangkaian Penaik Tegangan PWM

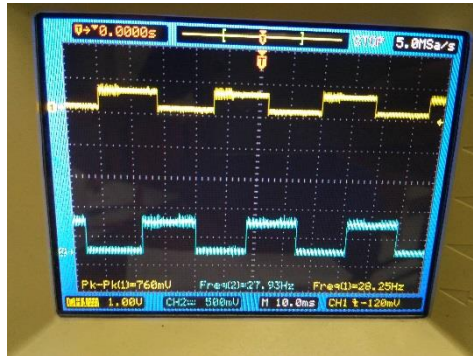
4.7 Pengujian Motor BLDC

Pengujian motor pertama dilakukan menggunakan PWM dengan frekuensi 500Hz sampai dengan 500KHz. Semua pengujian menggunakan PWM dengan Duty-Cycle 100% dengan tujuan untuk melihat respon gerak motor BLDC dengan jelas. Dari pengujian frekuensi yang telah dilakukan semakin besar frekuensi maka gerak motor akan semakin halus, sehingga digunakan frekuensi PWM sebesar 500 KHz karena transistor yang digunakan adalah TIP32 memiliki karakteristik batas frekuensi kerja maksimal adalah 1MHz. Dengan menggunakan frekuensi PWM sebesar 500 KHz maka nilai maksimal PWM berubah, semula dari 0-255 menjadi 0-144.

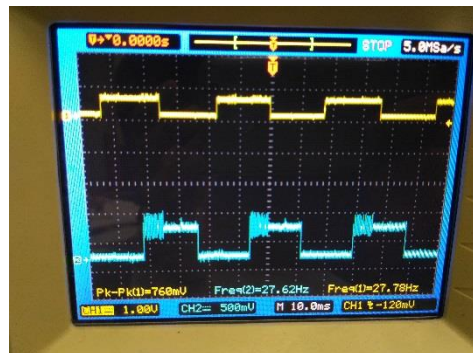
Tabel 4.2 Perbandingan Frekuensi PWM Pada Gerak Motor

No.	Frekuensi PWM	Tegangan Kerja Minimal	Keterangan Pergerakan Motor
1.	500 Hz	1.35V	Kasar
2.	5 KHz	1.40V	Sedikit kasar
3.	10 KHz	1.41V	Sedikit Halus
4.	100 KHz	1.35V	Halus
5.	500 KHz	1.38V	Sangat Halus

Pengujian kedua dilakukan terhadap kabel *hall sensor* pada motor. Sinyal dari masing – masing kabel diukur menggunakan *oscilloscope* dan didapatkan hasil sinyal dari tiap masing – masing kabel (U, V, W) sesuai dengan teori.



Gambar 4.26 Sinyal *Hall Sensor* Kabel U (atas) dan Kabel V (bawah)



Gambar 4.27 Sinyal *Hall Sensor* Kabel V (atas) dan Kabel W (bawah)



Gambar 4.28 Sinyal *Hall Sensor* Kabel W (atas) dan Kabel U (bawah)

BAB V

PENUTUP

5.1 Kesimpulan

Kesimpulan yang bisa diambil oleh penulis dari tugas akhir ini adalah :

1. Motor akan berputar kedepan jika sudut *pitch* terhadap *chassis* bernilai +, dan berputar kebelakang jika sudut *pitch* bernilai -.
2. Sistem penyeimbang otomatis pada *mini segway* roda dua belum mampu menjaga kestabilan pada sudut 0° terhadap *chassis*.
3. Adanya *death zone* yang terlalu besar pada *mini segway* yaitu sekitar 6.65° , antara -3.18° (posisi *chassis* maju ke belakang) sampai dengan 3.43° (posisi *chassis* maju ke depan).
4. Dapat melakukan pergerakan (*steering*) sesuai dengan perintah pengguna baik bergerak maju, mundur, belok kanan, belok kiri, dan berotasi dengan nilai parameter $K_p = 11$, $K_d = 3$, PWM maksimal untuk *chassis* bagian kanan sebesar 45, dan PWM maksimal untuk *chassis* bagian kiri sebesar 70 pada kondisi diberikan beban sebesar 1.5 Kg.
5. Sistem penyeimbang pada *mini segway* merupakan sistem non-linier.

5.2 Saran

Beberapa saran yang penulis bisa berikan untuk pengembangan tugas akhir adalah :

1. Penggantian *driver* motor *controller* BLDC dengan menggunakan IRAM, khususnya IRAMS10UP60B.
2. Penambahan komponen *enable* pengaktifan *mini segway* untuk pengamanan ketika digunakan. Bisa berupa penambahan *optocoupler* pada pijakan sehingga ketika dinyalakan *mini segway* tidak langsung seimbang, namun akan seimbang apabila baru dikendari/dipijak oleh pengguna.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Grasser, Felix; D'arrigo, Aldo; Colombi, Silvio; Rufer, Alfred (2001), "JOE: A Mobile, Inverted Pendulum", *IEEE Transactions on Industrial Electronics*, Vol. 49, No.1, pp.107 - 114, 2002.
- [2] Koutroulis, E., dkk., "High – Frequency Pulse Width Modulation Impelementation using FPGA and CPLD ICs", *journal of Systems Architecture* 42, pp. 332 – 344, 2006.
- [3] Bakhtiar Ramadhan, Eko Henfri Binugroho, Raden Sanggar Dewanto, Dadet Pramadihanto, "PENS-Wheel (Self Balancing One-Wheel Vehicle) Mechanical Design and Sensor System", *Proc. IEEE International Electronics Symposium (IES)*, pp. 442-447, September 2016.
- [4] Desna Riattama, Eko Henfri Binugroho, Raden Sanggar Dewanto, Dadet Pramadihanto, "PENS-Wheel (One-Wheeled Self-Balancing Vehicle) Balancing Control using PID Controller", *Proc. IEEE International Electronics Symposium (IES)*, pp. 31-36, September 2016.
- [5] Handry Khoswanto, Djoko Purwanto, "Keseimbangan Robot Beroda Dua Menggunakan Metode Fuzzy Logic", M.S. thesis, Dept. Electrical Engineering, University of Kristen Petra, Surabaya, Indonesia, 2010.
- [6] Michael Baloh, Michael Parent; "Modeling and model Verification of an Intelligent Self-Balancing Two Wheeled Vehicle for an Autonomous Systems", Desember, 15 2003 at Singapore.
- [7] WANG Xin, CHEN Songlin, CHEN Ting and YANG Baoqing, "Study on control design of two-wheeled self-balancing robot based on ADRC", *Proceedings of the 35th Chinese Control Conference*, Chengdu, 27-29 July, pp. 6227-6232, July 2016.

Halaman ini sengaja dikosongkan

LAMPIRAN

Program inisialisasi PWM pada Maple Mini :

```
//Inisialisasi PWM_1 (T2C1,pin11)
pwmtimer_1.pause();
pwmtimer_1.setPrescaleFactor(1);
pwmtimer_1.setCount(0);
pwmtimer_1.setOverflow(144);    //frek 500KHz
pwmtimer_1.setCompare(TIMER_CH1, 144);
pwmtimer_1.refresh();
pwmtimer_1.resume();

//Inisialisasi PWM_2 (T1C1,pin27)
pwmtimer_2.pause();
pwmtimer_2.setPrescaleFactor(1);
pwmtimer_2.setCount(0);
pwmtimer_2.setOverflow(144);    //frek 500KHz
pwmtimer_2.setCompare(TIMER_CH1, 144);
pwmtimer_2.refresh();
pwmtimer_2.resume();

//Inisialisasi pinMode
pinMode(BOARD_LED_PIN, OUTPUT);
pinMode(11, PWM);
pinMode(27, PWM);
```


Program PID :

```
//PID CONTROLLER
//Calculate Error
error_2 = setpoint_2-angle_2_FLFL;

//Integral
errorI_2 = errorI_2+(Ki*error_2*Tc);

//Derivative
errorD_2 = Kd*(error_2-error_sblm_2)/Tc;
error_sblm_2 = error_2;

//Penjumlahan Error
Sum_2 = error_2 + errorI_2 + errorD_2;

//Output PID
output_PID_2 = (int) (Kp*Sum_2);
```

BIODATA PENULIS



Civic Duta Mahendra lahir di Magelang pada tanggal 5 Juni 1995, merupakan anak kedua dari tiga bersaudara dari pasangan Bakri dan Ika Mahwayati. Penulis menyelesaikan pendidikan dasar di SDN 1 Salaman dan dilanjutkan dengan pendidikan menengah di SMPN 1 Salaman serta berlanjut dengan pendidikan atas di SMAN 1 Purworejo. Pada tahun 2013, penulis memulai pendidikan di Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Selama kuliah, penulis melakukan beberapa kegiatan baik dalam bidang sosial, *event/acara*, dan pengembangan diri berupa LKMM (Latihan Kepemimpinan Management Mahasiswa) dari ITS. Penulis mengambil Bidang Studi Elektronika sebagai fokus studinya. Penulis juga aktif sebagai Asisten Laboratorium Elektronika Dasar.

Email :

civiceduta@gmail.com

Halaman ini sengaja dikosongkan